2001-09

Modeling conventional land combat in a multi-agent system using generalization of the different combat entities and combat operations

Mert, Esref

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

MODELING CONVENTIONAL LAND COMBAT IN A MULTI-AGENT SYSTEM USING GENERALIZATION OF THE DIFFERENT COMBAT ENTITIES AND COMBAT OPERATIONS

by

Esref Mert
and
Erik W. Jilson

September 2001

Thesis Co-Advisor:	John Hiles
Thesis Advisor:	Rudolph Darken
Second Reader:	Michael Van Putte

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2001	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE
Modeling Conventional Land Combat in a Multi-Agent System Using Generalizations of the Different Combat Entities and Combat Operations

5. FUNDING NUMBERS

6. AUTHOR(S)
Jilson, Erik W. and Mert, Esref

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*maximum 200 words*)

There are inherent similarities between combat entities and between combat operations. In combat entities there exist common characteristics such as the ability to move, shoot, communicate and more. The level at which each entity is able to operate for these characteristics differentiates it from the others. For combat operations, a common characteristic is that all operations have a starting point, an objective point and an endpoint. The different operations take on unique properties based on where these points are located and what entities do at these points.

The generalization of the similarities in combat entities and combat operations can provide a framework that can assist developers and users to model the majority of combat situations with a single simulation. This thesis uses three different Multi-Agent System (MAS) combat models to demonstrate the generalization framework. Of the three models used, two existed previously and one was developed by the authors. Map Aware Non-uniform Automata (MANA) developed for the New Zealand Army and Defence Force and Archimedes developed by Least Squares Software LLC are the two existing models used. The model that was developed is based on the redesign of GIAgent developed by Captain Joel Pawloski, USA as a thesis at the Naval Postgraduate School.

14. SUBJECT TERMS
Multi-Agent System, Agent, Agent-Based Modeling, Agent Based Simulation, MAS, Conventional Ground Combat, Generalization of Combat Entities and Operations, Simulation, Combat Modeling

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

MODELING CONVENTIONAL LAND COMBAT IN A MULTI-AGENT SYSTEM USING GENERALIZATION OF THE DIFFERENT COMBAT ENTITIES AND COMBAT OPERATIONS

Esref Mert
1st Lt., Turkish Army
Turkish Army Academy, 1996

Erik W. Jilson
Captain, United States Marine Corps
B.S., United States Naval Academy, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN
MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION

from the

NAVAL POSTGRADUATE SCHOOL
September 2001

Author: [redacted]

Erik W. Jilson

[redacted]

Esref Mert

Approved by: [redacted]

Rudolph Darken, Thesis Advisor

[redacted]

John Hiles, Thesis Co-Advisor

[redacted]

Michael Van Putte, Second Reader

[redacted]

Rudolph Darken, Academic Associate
Modeling, Virtual Environments, and Simulation Academic Group

[redacted]

Michael Zyda, Chair
Modeling, Virtual Environments, and Simulation Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

There are inherent similarities between the numerous ground combat entities and the numerous ground combat operations. In combat entities there exist common characteristics such as the ability to move, shoot, communicate and more. The levels that each entity is able to operate for these characteristics differentiate it from the others. For combat operations, a common characteristic is that all operations have a starting point, objective point and an endpoint. The different operations take on unique properties based on where these points are located, actions enroute to points and what entities do at these points.

The generalized concepts in combat entities and combat operations provide a framework that can assist developers and users to model the majority of combat situations with a single simulation. This thesis uses three different Multi-Agent System (MAS) combat models to illustrate the generalization framework. Of the three "test" models used, two existed previously and one was developed. The two existing models are Map Aware Non-uniform Automata (MANA), developed for the New Zealand Army and Defense Force, and Archimedes developed by Least Squares Software LLC. The model (GENAgent) that was developed based on the redesign of GIAgent, developed by Captain Joel Pawloski, USA, as a thesis at the Naval Postgraduate School.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS

AAS	-	Adaptive Agent Simulation
AI	-	Artificial Life
CAS	-	Complex Adaptive System
CNA	-	Center for Naval Analysis
DoD	-	Department of Defense
EINSTien	-	Enhanced ISAAC Neural Simulation Toolkit
FM	-	Field Manual
GUI	-	Graphical User Interface
ISAAC	-	Irreducible Semi-Autonomous Adaptive Combat
JWARS	-	Joint Warfare System
LD	-	Line of Departure
M&S	-	Modeling and Simulation
MAS	-	Multi-Agent System
MCCDC	-	Marine Corps Combat Development Command
MDMP	-	Military Decision-Making Process
NAI	-	Named Area of Interest
NPS	-	Naval Postgraduate School
ONR	-	Office of Naval Research
RELATE	-	Relationships, Environment, Laws, Agents, Things, and Effectors

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

> The battlefield is a scene of constant chaos. The winner will be the one that best controls that chaos, both his own and that of his enemy.
>
> **-Napoleon**

A. MOTIVATION

"One of the best ways to model high-resolution ground combat is through the use of Multi-Agent Systems" (Ilachinski, 1997). The existing combat models that utilize Multi-Agent Systems (MAS) generally fall into two categories. The first group uses homogeneous forces that are capable of various types of capture the flag operations. The other group allows for unlimited range of force mixtures and operations but requires the use of a high-level computer language, like script languages. These simulations either restrict model building or overwhelm the users. Developers are forced to build situation specific simulations or high-level language simulations.

In "capture the flag" models, opposing forces are built with varying capabilities. Each force is placed into the simulation, with all of the members of that force having the same capabilities. These agents are typically considered cognitively "light-weight" or reactive agents (Weiss, 1999). This is different from today's armed forces, which must operate in task forces, coalition forces, and joint operations where unit and equipment capabilities vary greatly. All of these different force and equipment mixtures create the need for simulations that can handle multiple forces with varying equipment and capabilities.

The centralized emergent behaviors that are observed in MAS from the interactions of agents provide vital information for decision makers. Viewing this pattern of behavior in agents in a game of capture the flag requires intensive interpolation to carry over to an operation that might involve the planning of an assault on a fortified position. If the agents in the simulation were actually conducting an assault on a fortified position, the emergent behavior would be all the more easily interpolated and insightful.

In the “high-level language” simulations, many of the shortcomings of the “capture the flag” simulations, like homogeneous forces or executing different missions, can be solved. These agents are typically considered cognitively “heavy-weight” or “cognitive agents” with much richer behavior (Weiss, 1999). The difficulty lies in the need for the to learn the high-level language or to undertake the daunting task of developing a simulation using a high-level language. Additionally, these simulations tend to have scripted behaviors that cause them to be brittle, and fail in unforeseen circumstances.

So there exists a need for a MAS simulation that produces models that are easy to develop and use and that also provides model users with flexibility and control over agent behavior. This can be done through the development of a framework that generalizes the different combat entities and combat operations in a MAS.

B. THESIS GOALS

The main goals for this thesis are:

- Using the minimum number of characteristics, develop a generalization to describe simple ground combat entities. This generalization will allow one agent object, with different values for its characteristics, to describe any ground combat entity, from an infantry soldier to a machine gunner or to a tank.

- Determine the intrinsic and general concepts of simple ground combat operations. Using these generalities as goals, find rules that when matched to a goal could produce any simulated ground combat mission from an ambush to any other operation.

- Utilize an existing MAS simulation laboratory to demonstrate the successful implementation of these generalizations.

- Develop a MAS simulation laboratory to demonstrate and measure the successful implementation of these generalizations.

- Demonstrate model usefulness and potential, through the output and analysis of data. The data will be a combination of statistical quantifiable output and emerging qualitative behavior during the simulation. The results for each goal are summarized in the conclusion on page 76.

C. THESIS ORGANIZATION

This thesis is organized into the following chapters:

- Chapter I: Introduction. Identifies the motivation, goals, and organization for this thesis.

- Chapter II: Background. Introduces key concepts and terms. Argues the need for High Resolution Agent simulations. Examines existing High Resolution Agent simulations.

- Chapter III: Combat Entities. Identifies what generalizations are needed to describe combat entities.

- Chapter IV: Combat Operations. Identifies what characteristics are needed to describe the many different types of military ground combat operations.

- Chapter V: Development of Test Model. Describes the process and integration of the generalizations into existing MAS. Provides the steps taken to develop a simulation based on the generalization framework.

- Chapter VI: Scenarios and Experiments. Shows the ability for simulation laboratories using the right generalizations to describe many of the different scenarios existing in combat. Analyzes results from scenarios to show the operations specific pertinent data is produced.

- Chapter VII: Conclusion. Discuses the importance of having a laboratory where many different scenarios can be simulated. Points out the potential for further work in mission analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

> Often there is a gap between principles and actual events that cannot always be bridged by a succession of logical deductions.
>
> **-Clausewitz**

A. GENERAL

It is the goal of every commander to prepare their troops and their own leadership abilities for the fog of war (Clausewitz, 1943). Drilling forces in garrison or in the field prepares them for real combat. Training in some conditions or to a level of proficiency is not always feasible due to constraints like safety, security, space restrictions, and money.

Leaders also need to prepare for war. It is difficult to teach leaders how to effectively utilize soldiers and weapons in combat. Methods are needed to assist decision makers during training and combat. In training, these methods primarily consist of sand table exercises, group discussions, and war games. The decision makers can get assistance on the battlefield from intelligence sources and the opinions of their staff. Within the last 50 years, the use of computer simulations has become increasingly more important in the Military Decision-Making Process (MDMP). Simulations are the representations of real world events. They allow the exploration and examination of events without having to actually create or recreate the events

Simulation of combat operations contributes useful insights for many military decision problems (Hartman, Parry, and Caldwell). In order to achieve this insight and to represent the intended warfare, these models need to be accurate and believable. For military leaders, computer simulations provide information that is not always available when using sand table exercises, group discussions, or war games. These computer simulations do not provide answers as to who will win with absolute certainty. They can provide leaders information to assist in the decision making process.

B. KEY CONCEPTS AND TERMS

1. Multi-Agent System

Multi-Agent systems are systems in which several interacting, intelligent agents pursue some set of goals or perform some set of tasks (Weiss, 1999). Ferber (1999) defines the multi-agent system as a system comprised of an environment, a set of objects situated in the environment, an assembly of agents as active entities of the system, an assembly of relations between the agents, and an assembly of operations performed by the agents.

Combat simulations that use a multi-agent system approach try to explore combat as a self-organized emergent phenomenon, and take a bottom-up, synthesis approach, vice the more traditional top-down or reductionist approach.

2. Adaptive Agents

Adaptive agent simulations consist of a software environment, and numerous computer programmed entities (agents) that are situated on this environment. The adaptive agents can sense their environments, act upon what they sense, and adapt to the environmental changes and the actions of other agents. Additionally the environment may evolve/adapt over time due to the actions agents take on the environment. Chris Langton, who is considered by many to be the founder of Artificial Life, first demonstrated adaptive agent theory in 1995. His team at the Santa Fe Institute developed the simulation program called SWARM, a domain-independent multi-agent, discrete event simulation (Minar, Burkhart, Langton, and Askenazi, 1996).

Adaptive agent simulations (AASs) are often used in complexity research to study issues that are too complex to address any other way. They are routinely used to model life and other complex, non-linear systems. Axelrod (1997) also points out that an agent-based model is often the only viable way to study populations of agents who are complex in manner and adaptive, rather than fully rational. There is a critical distinction to be drawn between conventional simulations and AASs insofar as the AASs have no central controller. In AASs, individual agents self organize into larger units, which are themselves agents.

3. Complex System

In recent years there has been a rapid growth in the interdisciplinary field popularly known as the *Science of Complexity,* which studies the behaviors of complex systems. A complex system is dynamically composed of many non-linearly interacting elements (e.g., a society, an economy, warfare, or the human immune system). Complex elements exhibit behaviors in groups that they may not exhibit individually, and the behaviors are seldom predictable and may change over time. Interactions among the elements will change the system over time, and such changes cannot be attributed to any single rule or explanation.

Two features of the complex systems that make them complex in the common sense of the term are the non-linearity of the system and the type of interactions among the elements in the system. The combination of non-linearity, the large number of elements, and how these elements are "connected" and interact in the system, give complex systems the wide variety of phenomena. (Upton, 1998)

Complex systems can be further sub-divided into systems that are adaptive and non-adaptive. Adaptive systems use feedback to react and "learn", allowing them to adapt to their environment. Non-adaptive systems are complex primarily because of their interactions with other elements with no "learning" or adapting. The reason for focusing on complex adaptive system in simulation is that it provides insight into social systems and human political constructs. Any group of humans who interact will, over time, form a unique system broadly similar to those researched in complex adaptive systems or agent programs. Just like in complex adaptive system theory, humans build all sorts of social structures and engage in complex behaviors. Such structures create their own rules, and are thus fundamentally unpredictable. (Bassford, 1998)

4. Combat Simulations

Since the beginning of computer simulations, military analysts and programmers have developed different combat simulations for every scale of combat operations. F.W. Lanchester introduced a set of coupled ordinary differential equations – now commonly called the Lanchester Equations (LEs)- in 1914, as models of attrition in modern warfare (Lanchester, 1995). The LEs have since served as the fundamental mathematical models upon which most modern theories of combat attrition are based.

Some commanders can't accept uncertainty on the battlefield, believing success would lie with who ever has the most information. But today, it is regarded that military conflicts, particularly land combat, posses all the characteristics of uncertainty and complex systems (Ilachinsky, 1996). Combat simulations are more and more frequently modeling land combat as a complex adaptive system. This leads to the exploration toward alternative, non-Lanchesterian descriptions of combat.

Combat simulations have been developed either as low-resolution or high-resolution simulations. The resolution depends on the scale of the combat and the size of the smallest force to be modeled.

5. High Resolution Combat Simulations

High-resolution combat models, which are detailed models of warfare, represent individual combatants as separate entities with numerous attributes. In these models the combat process is broken down into high-resolution sequences of events and activities. The main goal is to model each combat phenomenon so that results are traceable to specific physical data or to specific behavioral assumptions. High-resolution land combat simulations are usually not developed above the battalion level. Depending on specific scenarios, they create a synthetic combat environment by providing a credible representation of the battlefield, including physical, behavioral, and environmental models. Most high-resolution combat simulations are traditional AI (Artificial Intelligence) based. Typically this approach applies non-adaptive, "if-then" rules. In recent years the battlefield is being modeled more often as High-Resolution Agent-Based Combat Simulations.

6. Low Resolution (Aggregated) Combat Simulations

In low-resolution combat models, individual combatants are aggregated into larger units. The entities represent groups rather than individual combatants. This approach helps the large-scale combat modeler decrease the number of simulation entities to a manageable number by sacrificing detail for scope. However, by the nature of this approach, information about individual differences is lost, modelers lose track of what each individual is doing at a given time, and the information about event sequencing is lost since the simulation does not keep track of individual actions. These aggregate models lack the ability to represent battlefield complexity and the complex relationships that exists among the entities on the battlefield. Like most existing models, most low-resolution combat models remain Lanchesterian in nature, the driving factor being force-on-force attrition. We believe that these models of land warfare are insufficient for assessing the advanced warfighting concepts. They homogenize the properties of entire populations and ignore the spatial component altogether.

C. LAND COMBAT SIMULATIONS

1. Low Resolution vs. High Resolution Combat Simulations

As Davis (1993) expressed in his study, after defining concepts of different levels of resolution and their applications to combat modeling, we can classify the uses of low and high-resolution models.

Low-resolution models are mostly used for broadband or “big picture” comprehension, system and policy analysis, decision support, adaptability, low cost and rapid analysis, and making use of low-resolution knowledge and data.

High-resolution models are mostly needed for understanding behavior phenomena, representing knowledge, simulating reality, calibrating or informing lower resolution models, and making use of high-resolution knowledge and data.

2. High-Resolution Agent-Based (Adaptive) Combat Simulations

Today's vision of combat can be expressed as small, highly trained, well-armed autonomous teams working in concert, continually adapting to changing conditions and environments. To address shortcomings in existing conventional models, programmers are exploring developments in complex systems theory and Multi-Agent System (MAS) simulations. Since high-resolution agent-based land combat simulations are developed to model the emergent phenomena resulting from the complex, collective, nonlinear, decentralized interactions among notional combatants, they take a bottom-up approach to the modeling of combat, vice the low-resolution or aggregated models' top-down or reductionist approach.

As Ilachinsky (1996) expressed in his thesis, land combat can be best modeled as a complex adaptive system. This theory led to the exploration of complex, interactive behaviors of combat by using self-adaptive, MAS simulations. The motivation behind this study is to explore alternative non-Lanchesterian description of combat.

High-resolution, agent-based combat simulations can be regarded as an interactive conceptual laboratory for identifying, exploring, and possibly exploiting self-organized, emergent collective patterns of behavior in combat.

D. SURVEY OF SIMILAR HIGH RESOLUTION, COMBAT, MULTI-AGENT BASED SIMULATIONS

1. ISAAC

Developed by Dr. Andrew Ilachinski in 1997, ISAAC (Irreducible Semi-Autonomous Adaptive Combat) is one of the first military research projects to attempt to model land combat using agent-based simulation techniques. It is sponsored by CNA (Center for Naval Analysis) and ONR (Office of Naval Research). The goal was to take a bottom-up approach to the modeling of combat, vice the more traditional top-down, or reductionist view. Basic elements of the simulation are ISAACAs (ISAAC Agent) representing the primitive combat units (infantryman, tank etc.), and the battlefield which is represented by a two dimensional lattice of discrete sites. ISAAC can be considered as the first phase "proof-of-concept" multi-agent based simulation of land combat. Detailed information can be found in Ilachinski's own thesis paper (Ilachinski, 1997).

2. EINSTein

EINSTein (Enhanced ISAAC Neural Simulation Toolkit) is considered as the second phase in the development of multi-agent based land combat simulation incorporating much more functionality with its windows-based development environment. It is an ambitious follow-on project that takes lessons learned from ISAAC and is regarded as self-contained "laboratory" for exploring evolution of self-organized collective behavior from primitive local dynamics in battlefield.

Some features of EINSTein include it's fully integrated Windows 95 GUI front-end, an object-oriented C++ code base, context-dependent and user-defined agent behaviors (i.e., personality scripts), on-line genetic algorithm, neural-net, reinforcement learning, pattern recognition toolkits, data collection, multi-dimensional visualization tools, on-line chaos-data/time-series analysis tools, and on-line mission-fitness co-evolutionary landscape profiles.

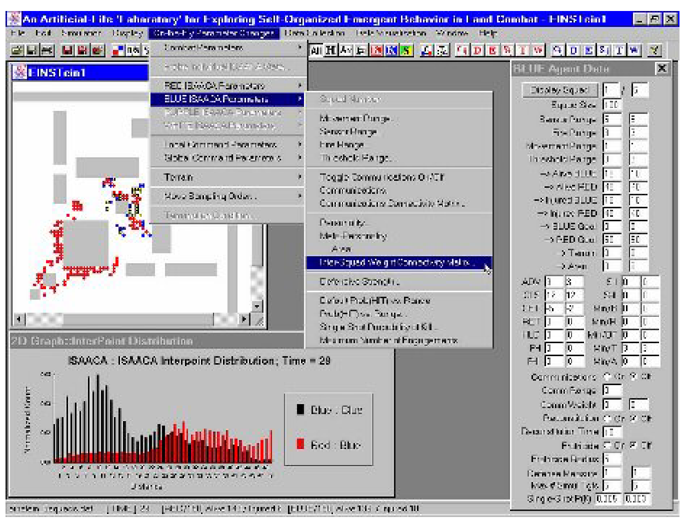


Figure 1   Sample screen snapshot of EINSTein

3. Naval Postgraduate School Agent Based Simulations

After the development of the first agent based adaptive combat simulations, an increasing interest has been aroused in this area. There has been also an intensive focus at the Naval Postgraduate School in developing different aspects of combat simulations using agent based modeling following ISAAC. Unrath (2000) developed a helicopter reconnaissance simulation using the agent-based approach. It was intended to be a simulation laboratory used in the acquisition cycle to examine support planning for the Comanche helicopter. Dickson and Roddy (2000) developed an agent-based land combat simulations called JACOB (namely Son of ISAAC) as a sample application of their RELATE architecture in Java. RELATE is a MAS architecture focusing on six key concepts; relationships, environment, laws, agents, things, and effectors. The authors achieved outstanding success in their research and their work has been the foundation for further research in the area of land combat modeling. Pawloski (2001) used the RELATE architecture to create GIAgent. It is a MAS simulation tool developed to examine the relationship between maneuver and unit organization.

4. Archimedes

Least Squares Software LLC of Albuquerque, NM developed Archimedes for the Marine Corps Combat Development Command (MCCDC). Archimedes is currently in its beta version. It is a tailorable agent based modeling platform. It provides the ability for the user to create agents and build terrain. To help define the agents, Archimedes allows aspect variables or characteristics, connections or inter-agent relationships, and rules for the agents and connections to be created. Once all of the parts have been defined a scenario can be created that inserts agents with connections onto the simulation terrain.

Conceptually, the brain of the agent is modeled separately from its body. The algorithms that perform specific functions for specific purposes are located in the aspects, which are software objects that contain algorithms for functions of the model such as communicating, firing, moving, detecting, etc. Aspects can be regarded as the workings of the body of the agents. Variables are defined as "fuzzy" variables (i.e. for a “discipline” variable: High, Medium, Low as real world expressions) instead of quantifiable values or Booleans etc. This provides user with flexibility in developing the models or scenario (Reynolds and Dixon, 2001).

5. SWARM

SWARM was developed at the Santa Fe Institute in the mid 1990’s, with a beta released in 1996. Swarm is a software package for multi-agent simulation of complex systems. Swarm is not specifically designed for combat simulations, but as a tool for exploring a wide variety of complex systems. The base group or swarm is a collection of communicating agents. Swarm allows nested structures where one agent can be made up of a swarm. It requires a C compiler, Unix, and X windows. (Minar, Burkhart, Langton, and Askenazi, 1996)

6. SOAR

Soar is a general cognitive architecture for developing systems that exhibit intelligent behavior. Researchers all over the world, both from the fields of artificial intelligence and cognitive science, are using Soar for a variety of tasks. It has been in use since 1983, evolving through many different versions to where it is now Soar, Version 8.2. (Rosenbloom, Laird, and Newell, 1993)

7. JANUS

JANUS is a combat simulation originally developed at the Lawrence Livermore National Laboratory. The JANUS simulation is an interactive, high-resolution model of ground combat at the entity level. In most configurations JANUS requires some degree of contract civilian support staff to operate and maintain the simulation. The time to configure JANUS can be significant. It could take a week to load a brigade size unit’s data and place them in the simulation with initial orders (TRAC, 1999). JANUS is a Semi-Autonomous simulation, meaning that unit missions and operations are planned and executed by human users, and the software performs individual entity reactive actions autonomously.

8. MANA

MANA was developed for the New Zealand Army and Defense Force and is also being used with the U.S. Marine Corps initiative, Project Albert. Roger Stephen and Michael Lauren developed the software. MANA is currently in its beta version. MANA is very similar to the earlier works such as ISAAC/EINStein. It is a MAS that's first use is as a bottom-up abstraction of the essence of a scenario. Along with the similar features that ISAAC has, MANA also has situational awareness for intra-squad communications, a terrain map, and waypoints. The terrain map is just bitmap images and most bitmap editors can be used to create a map. In its present stage it has only two distinct colors as terrain features. Grey represents barriers, or impassible objects. Yellow is used as an "easy going" route. The waypoints are a set of points to follow on the way to the objective point.

E. SUMMARY

Of the above listed models only MANA and Archimedes have the flexibility to use our proposed generalization framework. The framework can also be demonstrated through the redesign of GIAgent. The redesign would be a complete code re-write using only the GUI interfaces and the base agent interface (RELATE).

III. COMBAT ENTITIES

A. GENERAL

The characteristics of the agents are the first things that need to be identified in combat MAS. Three abilities to describe these combat entities quickly come to mind. These are the ability to move, shoot and communicate. Each one of these can be described in great depths and using many different characteristics. These are examples of some of the tangible characteristics of combat entities. There are also intangible characteristics, which include things like attitude, discipline, obedience and motivation. The task of trying to model every characteristic of a combat entity is possible, but intractable. So the question that faces an analyst using an existing MAS or developers of new MAS is, “What characteristics should be included in modeling combat entities?” Of course, the answer is “just the right amount”. “Just the right amount” can be described as enough characteristics to gather valuable information and not too many that provide redundant or insignificant information.

For the scope of this thesis only ground combat entities and direct fire weapons will be addressed. Although there are common characteristics that exist between a tank and a naval ship, there are also many different characteristics that prevent an easy direct correlation.

There are also different types of combat units that need to be modeled. An infantry platoon is obviously different from a tank platoon, but they do have similar characteristics. Both of these examples move, shoot and communicate. The difference lies in a combat entities’ ability to perform each characteristic. So by adjusting the levels of different characteristics, a wide range of units can be modeled. Thus, a generalization of the essential characteristics of combat entities can provide an easy way describe many different combat units.

B. ENTITY CHARACTERISTICS

1. Movement

The first ability that we will address is the movement of combat entities. This ability can be described in great detail. It could include algorithms that adjust the speed of an agent based on terrain, restrict movement in certain areas of a battlespace, or simply described the maximum number of terrain boxes that can be moved during a time step.

All simulations need at least a max speed characteristic to describe movement. Defining the different speeds of the entities will allow proper ratio of movement between the entities and the terrain. A model could get very detailed on how terrain and agent's interactions affect movement speeds. These types of movement algorithms based on the environments can be expanded once the minimum of a reference speed exists.

2. Sensing Range

Agents must also be able to sense other agents and their environments. This is vital to enable entities to interact with the simulation. The sensing range involves visually or maybe electronically 'seeing' what is in an agent's environment. The ability to sense in a simulation can be simply based on straight distance or it may involve accounting for concealment, line of sight, smoke, or degradation due to night. Regardless of the complexity the simulation needs at least a basic sensing range.

3. Communication Range

An agent may not be able to "see" another agent, but the capability should exist to allow it to communicate up to and possibly beyond its sensing range. This type of communication would be important for both centralized and decentralized control of forces. This communication could be a type of radio transmission or just voice to a nearby entity.

4. Lethality

The ability to describe the "punch" a weapon possesses can be labeled as its lethality. This is another characteristic that is necessary to differentiate between a rifle and something like a tank's main gun. This might be a distance from impact that a weapon has a lethal effect or an amount of damage the impact does to an agent's durability levels. The ability to represent non-lethal weapons or weapons that are effective against harden targets could be represented with this characteristic.

5. Weapons Range

The maximum distance a certain weapon can be effective is important to model different weapon types. A weapons range may be a different distance than an entities sensing range. Although they could be the same, a sensing range will most likely be equal to or greater than an entities weapon range. A combat entity could have multiple weapons, but its weapons can be aggregated into one weapon in a distillation simulation. In a simulation where the details are not modeled a tanks main gun, its heavy and medium machine gun can be represented by an entity with one weapon. This weapon is what the entity would use against all entities it engages. Its effect on the other entities would be based in its lethality and the target or targets' durability.

6. Durability

In order for an agent to represent a soldier or a tank, one of the characteristics that are needed is a property that can be adjusted to reflect the agent's survivability. It may require only one shot from a rifle to kill a soldier, but many shots from a rifle will not kill a tank. Thus there must be a way to define an entities durability or health. This characteristic would be difficult to quantify; but having a scale from 0 to X could be used to create a ratio of survivability or durability.

A hard target would be modeled with a very large durability value and a soft target a small durability. Then an entity's lethality could be given a very large number and be directly related to the durability. In this case a tank would have a high lethality and up against another tank with a high durability a hit would create a kill or crippling damage. This would also allow a weapon with a small lethality to be virtually useless against a hard target.

7. Probability of hit

The probability of hit is how well a target can be hit by the shooter when engaged. The ability to adjust the probability of hit for an entity provides the ability to represent a bad or expert gunner and the effectiveness of sophisticated target acquisition systems. It can also represent a low level of training. Having a probability of hit characteristic might also be used to represent a weapon that might not be effective as others.

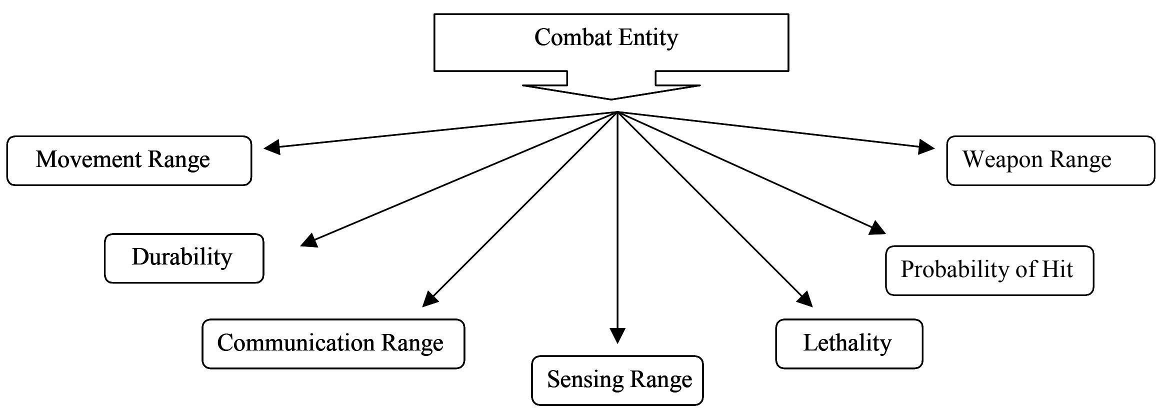


Figure 2 Generalization of Combat Entity

C. SUMMARY

While most of the characteristics listed above are tangible characteristics, it would be easy to add intangible characteristics. A common way intangible characteristics are incorporated into a model is to give it a name and based on an adjustable level, have it effect tangible characteristics. An example would be a training level characteristic, where if set to 75% then the probability of hit might be lowered by some percentage or linearly. There can be far more complicated implications of a low training level. Another example would be a motivation level that could be changed by lowering the break point for a unit with low motivation.

These characteristics should have the ability to be adjustable to a wide range of possible values. Allowing the speed to be increased passed a known threshold could be used to explore new technologies. This gives the user the ability to try concepts that might not have been evident to the developers. This ability to generalize also eliminates the need for something specific like a probability of kill table and lets the analyst determine how two platforms or entities match up based on its generalization characteristics. These characteristics are for the individual entity. To model units, the type of unit and the number of entities in that unit are also needed.

So the above characteristics can be thought of as a framework to assist analysts and developers generalize combat entities for a MAS combat simulation. The framework for the entities does not include all possible characteristics, but is the minimum needed to be able to describe all the different types of units. More characteristics can always be added to add a capability or answer a specific question. The end result is always to have a model that can provide insightful information to the questions being explored. With just these few characteristics, information can be provided to most of the questions being asked about an entity in a conventional ground combat scenario.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. COMBAT OPERATIONS

A. GENERAL

For the purpose of this paper the term “combat operations” is meant to express “tactical level land combat operations”. Like the many different types of combat units there are also many different types of combat operations. Military manuals describe the proper procedures for conducting these different operations. There are manuals for conducting raids, ambushes, mechanized assaults, and other operations. Different types of forces have different manuals for how they specifically carry out these operations. Each of these manuals explains the properties and different aspects of the operations and are very different from the each other. All these manuals are needed to understand and execute the complicated area of combat operations. Modeling combat operations in MAS could be made easier with a framework that generalizes combat operations.

A developer of a combat simulation must develop a simulation that is either operation specific or general enough that it can handle different types of operations. If a model is operation specific, then an analyst could be restricted in what they are able to simulate. On the reverse side, an analyst could have the difficult task of learning how to use a simulation that is large and complicated.

Combat operations can be classified into three basic types; movement, offense, and defense (FM 7-8, 1992). While this is a valid way to categorize combat operations, it is too broad. There are special types of operations, like an ambush and raid that do not easily fit into these categories. An ambush has both offensive and defensive characteristics. In an ambush the enemy force is sought out, but once in position, the friendly force lies in wait. A generalized simulation must have more categories than just movement, offense and defense. The characteristics of offensive and defensive operations as described in U.S. Army Field Manual 3-0 are listed in Appendix A (FM 3-0, 2001).

The type of force conducting an operation can affect the way the operation is carried out, as well. A mechanized assault is very different from an infantry assault.

Like the similarities between a tank and a soldier there are also similarities in the way different forces carry out the same operation. A generalization of the different features and components of different combat operations would allow for an easier way to model and simulate each of them.

B. GENERALIZING DIFFERENT COMBAT OPERATIONS

Combat operations have their own characteristics, with different procedures and application. We can generalize them by utilizing the commonalities they possess. Whatever they are, either offensive or defensive, all combat operations can be described based on three main points and the actions at these points. Any number of waypoints can also be added between these three main points. The three points are a starting point from which to begin, an objective point where to execute the main operation, and an ending point where the mission is finished. These three points (starting point, objective point, and end point) will also be called "operational points" in this paper. See Figure 3 for a simple representation of the three operational points.

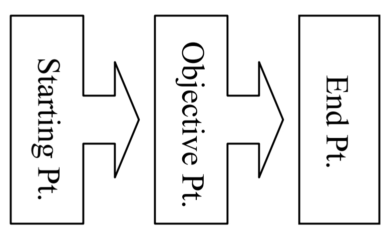


Figure 3 Three points of combat operations

A scenario could involve a unit that has multiple objective points. In this situation the scenario could be divided into two different simulations and then analyzed either by combining the data or separately. There could also exist a scenario that involves a long movement to or from an objective area. If there is no expected enemy contact during this movement, then this portion of the scenario could be left out and the starting point would begin at a point on the map where enemy contact is possible. If enemy contact is possible before or after an objective point, then here also the scenario could be divided up into two different simulations. The two simulations would involve a movement operation and an operation near the objective.

The first point that needs to be defined is the starting point. All operations must start from somewhere in the terrain. It may be a base, a position to defend or an assembly area. Conversely, if there is a starting point there must be an end point, a point when reached signifies mission completion. In between these two points is the objective point. This is where the planned action is to be conducted. These three points may all be in different locations, the same location or any combination in between. For example, an assault could have all three points in different locations (see Figure 4). This would represent a force traveling from an assembly point (starting point) to an enemy position (objective point) and then to a withdrawal position (ending point). The other force in Figure 4 is defending a position (all three points the same).

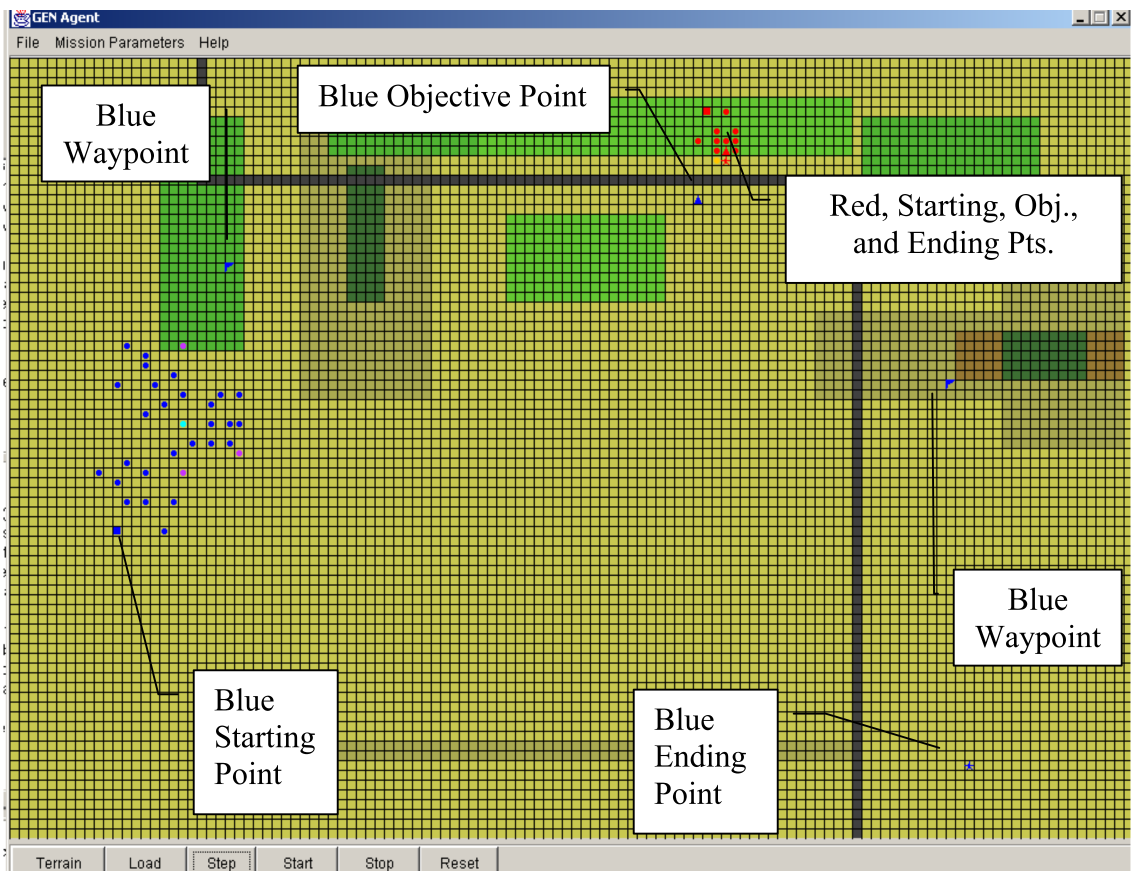


Figure 4   Blue Assault/Red Defense example

There can be many different variations using these points. There are also some further distinctions that can be made to add even more possibilities to the different combat operations. These further distinctions are what is done upon enemy contact while traveling between points and what is done at the objective point.

C. ACTION AT OBJECTIVE POINT

There are three basic actions at the objective point; make contact with the enemy, wait for enemy contact or end mission. The option to withdrawal based on an overwhelming force is not covered for this generalization. Nor are special operations or reconnaissance addressed in our generalization, thus all of these actions would involve the force engaging the enemy once it is sensed. The function of determining and making contact with the enemy is indicative of an attack or raid. Waiting for the enemy goes along with defensive operations or an ambush. The end mission action usually applies to movement operations. This is because a movement operation ends when a force reaches its objective. See Figure 5 for a sample graphical representation of possible actions at the objective point.

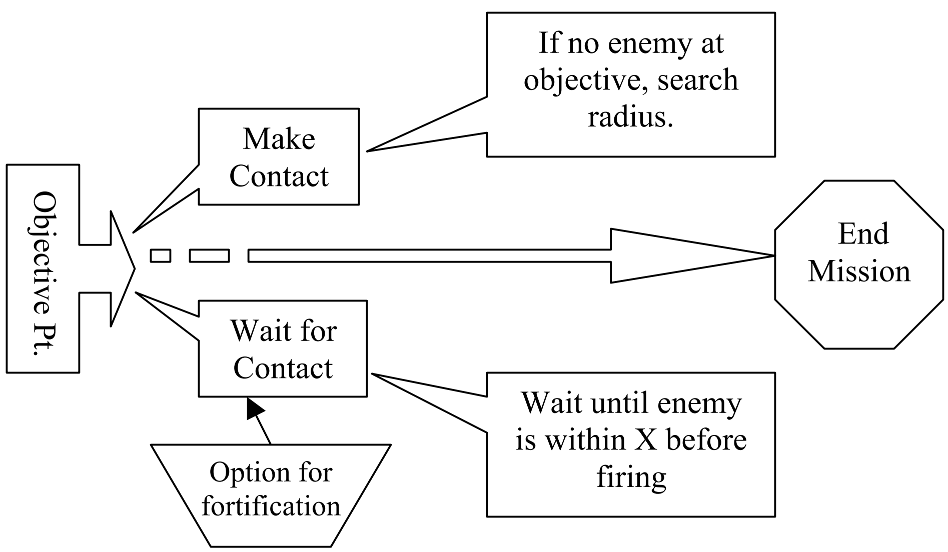


Figure 5 Action at objective point

For the “make contact” option, the objective point can be thought of as a point where enemy forces are expected. If a unit gets to an objective point and an enemy has not been engaged, the unit may actively seek out the enemy around this point. This seeking out the enemy should have a set pattern or a distance from the objective point to constrain the forces (like patrolling in the objective area).

If the action is to wait for contact, then the unit is going to prepare and wait for the enemy. Although this can be thought of as a type of defensive operation, it could also be applied to special operations, like an ambush. To distinguish between these two operations, the option of waiting and fortifying a position can be used. This would be more indicative of a defensive operation. A way to define this option might have a force’s durability increase up to a certain level over time. Laying in ambush would not generally involve extensive fortification of positions.

A mission could end if a force reaches its objective point. This could happen if the objective point and the endpoint are the same location for example in a convoy operation. Once this point is reached the convoy has conducted a successful mission, and thus the mission is over.

Once enemy contact is made, the forces must have a set of rules to determine in what way to react. Action upon enemy contact applies while traveling to different points and at the objective point. If we can generalize actions that can be taken at these times, it makes for a less complex set of instructions to program into different scenarios. Actions on contact while traveling are different from actions at the objective point.

D. ACTION UPON ENEMY CONTACT

An enemy force within sensing range of a unit becomes a new part of the MAS environment that must be dealt with. One way to generalize the possible courses of action when contact is made with the enemy is with the ability to attack, hold position, drawback, or push through - or keep going. The direction of movement of the force while engaging would be the variable for these actions.

See Figure 6 for a graphical representation of possible actions upon enemy contact.

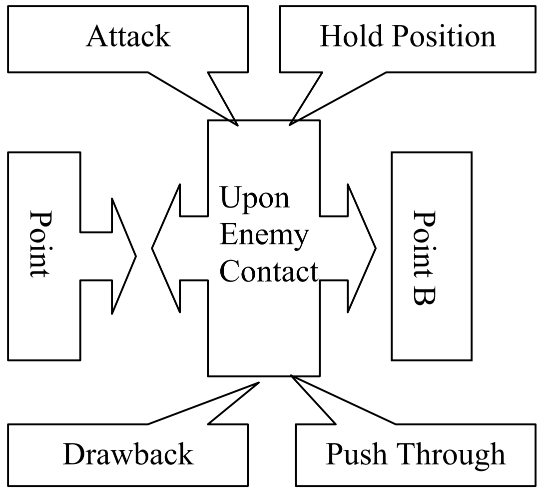


Figure 6 Action upon enemy contact between points

In an attack rule, the attacking force would move towards the detected enemy. A hold position action would involve firing weapons but no movement. The drawback option would be a form of retreat, where the force would retreat towards a previous waypoint or operational point. If reaching a point or objective was more important than engaging the enemy, then the forces' action would be to push through. The direction for the push through option would be towards the next waypoint or an operation point.

E. OPERATION TERMINATION

One of the most common ways to stop a simulation is based on time. In some simulations, regardless of what is happening with the agents, all action stops after a set time or number of time steps. This is an easy solution to terminating a simulation, but not always the most useful for collecting statistical outputs. A more realistic solution would be the ability to stop a simulation based on a force's "break point" or when a force reaches its end-mission point or when a force waits for enemy at a position more than a "time out" limit.

An important option in a simulation is the ability to experiment with a force's breakpoint or the percentage of loss at which point a force determines that further combat action would not be advantageous. In a scenario of an attack on a defensive position, the defensive forces' breakpoint would be that when they could no longer defend their position and have to retreat or surrender. This percentage of loss or "breakpoint" could easily be a variable that is set at the beginning of the simulation for both forces.

The three points of a combat operation also provide a base to terminate a simulation. In some operations when a force reaches its objective/end point, it is also the end of the operation. If the operation is a movement from one point to the next and if the force makes it to its objective/end point then it has conducted a successful operation. There is no need to run the simulation after the endpoint is reached.

The operation being modeled or aspect of a battle being explored will determine the type of operation termination method that is necessary. With the three options of ending a simulation, based on time, breakpoint, or reaching an endpoint, will provide a robust simulation that is still easy to understand and program.

F. SUMMARY

There are details of a scenario that will be lost in this generalization. An example would be a listening or observation post for a defensive position. The decision has to be made if these details are important to the vital part of a scenario that is being analyzed. The vital part being the main forces' defense of a position. If these details are important enough, than the simulation could be divided into two simulations. A simulation that models the main battle of the defensive force and a simulation that models only the interaction and detection of the listening or observation post against the opposition.

An operation can be described in more than one way with this framework. For example, a defensive operation may have all three operational points at the same location and a "wait for contact" characteristic. Or a defensive operation could have a unit moving from a starting point to an objective point where they will set up a defensive position and then "wait for contact". A list of some example settings for different operations is listed in Appendix B. This list does not include all the operations listed from the U.S. Army Field Manual 3-0 in Appendix A. The example settings are intended to be a guide for an analyst and not a limitation on what can be modeled. In a MAS all the details of some of the operations in FM 3-0 need not be included in order to gain insightful information from a simulated engagement. This generalization will work for most of the conventional ground combat missions. The generalization of combat with these points will aid in the development of a robust combat simulation. This framework will also provide much more analytic benefits than using a “capture the flag” scenario.

V. DEVELOPMENT OF TEST MODELS

A. GENERAL

We have chosen three different types of MAS modeling tools to demonstrate the validity of our ideas about the generalization of combat entities and combat operations. These simulations are distillation models, or models that do not use a high level of detail. The first modeling tool used is Map Aware Non-uniform Automata (MANA), developed for the New Zealand Army and Defense Force by Defense Operational Technology Support Establishment (DOTSE, 2001). The next model utilized is Archimedes developed by Least Squares Software LLC. Finally, we developed our own model based on extensive modifications of GIAgent, developed by Captain Joel Pawloski, USA, as part his thesis work at the Naval Postgraduate School.

MANA and Archimedes have many more capabilities that are in the generalization framework. In these tools the framework serves as a starting point to assist in the beginning stages of model building. Once the scenario is modeled in its basic form with the generalization framework, more features can be added depending on the analyst's needs. Conversely, GENAgent was done for the specific purpose of matching the generalization framework to combat operations.

B. MANA MODEL DEVELOPMENT

1. General

MANA was developed to create a complex adaptive system to examine combat. It is an original piece of software that builds on earlier works such as ISAAC/EINSTEIN and the evolving Archimedes model. The version of MANA used is a beta, revised in April of 2001. Version 1.0 of MANA is due in the summer of 2001. Some of the features that differentiate it from other MAS models are situational awareness, a terrain map, waypoints, and event-driven personality changes (DOTSE).

a. Situational Awareness

Agents in MANA have the option of sharing a collective picture of sensor information. A headquarters squad is assigned to each alliance and all squads with a headquarters squad for that alliance report their sensed environment to the headquarters squad. This headquarters squad then sends the information back to all agents in the alliance. This feature gives agents the ability to communicate enemy positions outside of an agent's local sensing range. MANA includes a map that can be viewed during a simulation run.

b. Terrain Map

The maps in MANA are just 350x350 pixel bitmap images. In the present version there are only two terrain features that the agents interact with, and they are represented and distinguished by colors. The first is a solid object that is impassable by agents. This barrier is represented in gray. The other terrain feature is paths or roads. These are represented in yellow and agents can be given the propensity to stay near or on the path.

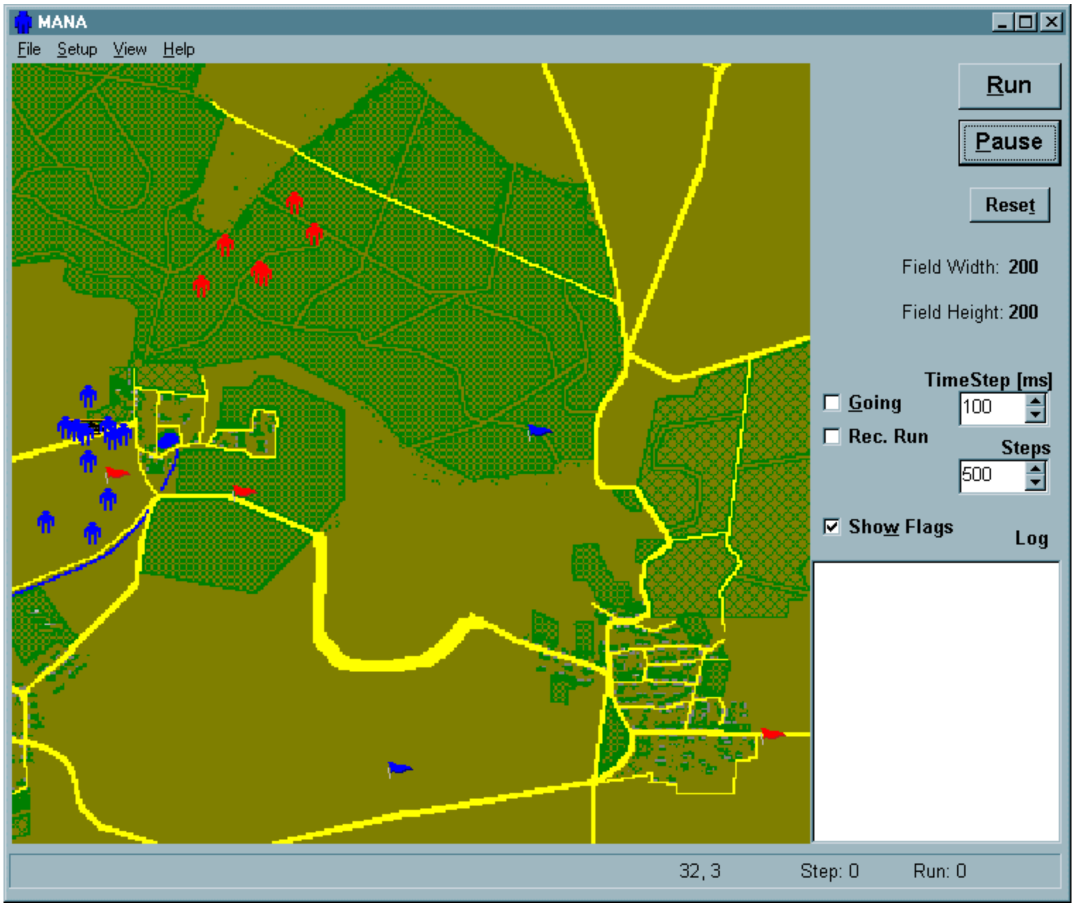


Figure 7 Sample Terrain Map

c. Waypoints

MANA allows for waypoint to be placed in the battlespace to guide the movement of agents. An agent has personality settings that can attract it to the next waypoint or repel it away from the waypoint. Waypoints are entered directly onto the terrain map in the general squad properties menu. The first point entered is point (0), which represents the final waypoint.

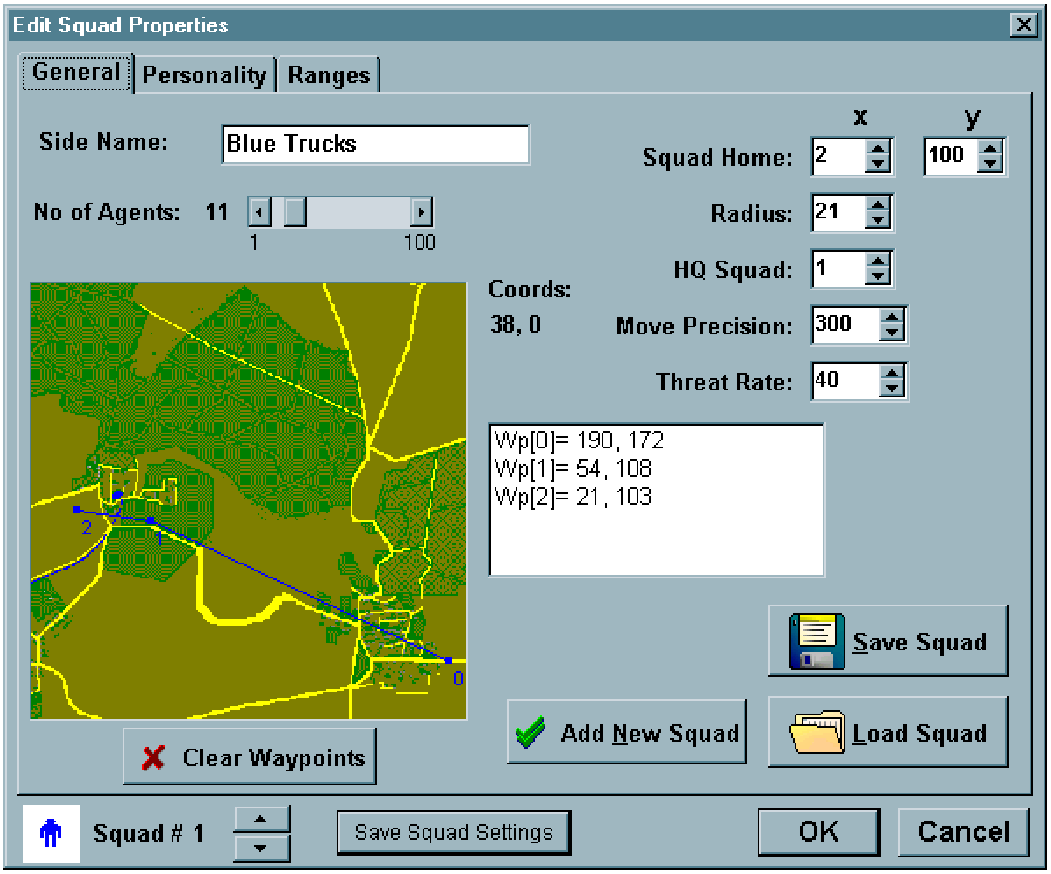


Figure 8 General Squad Properties Menu

d. Event-Driven Personality Changes

Agents have different personalities depending on their current state. If an event occurs that changes its current state, an agent's personality can change to reflect the settings in the new state. The state changes can affect allegiances and capabilities for properties like sensing and weapon range. These personalities can affect a specific agent or a squad of agents.

Personality Element	Description	Controls propensity to move toward/away from
w1	Alive Friends	Agents of same allegiance or just squad
w2	Alive Enemies	Agents of enemy allegiance

w3	Injured Friends	Injured agents of same allegiance or just squad
w4	Injured Enemies	Injured agents of enemy allegiance
w5	Next Waypoint	The next waypoint agent's squad has been assigned
w6	Enemy's Flag	The enemy's final goal (waypoint)
w7	Easy Terrain	Easily traversed terrain, i.e. roads
w8	Enemy Threat 1	Enemies in SA map which are of low threat
w9	Enemy Threat 2	Enemies in SA map which are of medium threat
w10	Enemy Threat 3	Enemies in SA map which are of high threat

Table 1 Personality Weightings

2. MANA and Generalization

In this section the features of MANA that only directly apply to the generalization will be discussed. The other parts are described in depth in the MANA Combat Model help file.

a. Combat Entities Generalization

MANA's architecture allows for easy adaptation of the generalization of entities. Like our framework it has a sensor range, a firing or weapon range, movement range or speed. MANA also has a "firepower" setting that represents a single shot kill probability or probability to hit as it is called in the framework. The durability of an agent can be modeled using MANA's "number of hits to kill" setting for each agent.

The lethality of a weapon does not have a direct relation to any setting in MANA, but there is a "max targets per step" setting that determines how many agents can be engaged at one time or the number of hits on one agent per step. This setting could be used as a type of weapon lethality. Thus allowing one agent to engage more than one enemy.

There is no communication range setting in MANA. Agents are able to report enemies in their sensing range to the HQ agent/squad, regardless of their distance to that HQ agent/squad. There is a value, “threat influence range” which represents the radius of the situational awareness map communicated. The HQ agent/squad sends a limited overall situational awareness to all agents in its alliance, which represents a communication of all enemy agents perceived by all friendly agents.

b. Combat Operation Generalization

MANA’s architecture allows for easy adaptation of the generalization of operations. The starting point can be set with the “location (x, y)” setting. This location is where the agents are initially placed. The ending point is the first waypoint that is designated. The objective point has no direct correlation in MANA. There is a state for reaching a waypoint. If one waypoint is defined, then that point can serve as both the endpoint and objective. In this case the actions at the “waypoint’ can be set appropriately. If two waypoints are defined, then one waypoint can act as an objective with the necessary actions and the other waypoint is the endpoint. In this case, the event of arriving near a “waypoint” will take place at both points. Thus, simulation needs to terminate when the agents reach the endpoint.

The “upon enemy contact” action of an operation can be directly related to a number of different trigger states in MANA. These states will be called “in contact” states. The “upon enemy contact” can be set on an agent’s or a squad’s event state of “taken shot”, “shot at”, or “enemy contact.” In order to represent an “attack”, once an agent’s state is changed to one of the “in contact” states, its propensity to move towards either alive and/or injured enemies can be increased. For an agent to “hold” its position, its movement speed can be set to zero when one of these “in contact” states is triggered. For the “drawback” and “push through” options for an agent, the attraction or repulsions to the next waypoint could be adjusted.

The action at the objective point will be based on the trigger state “next waypoint”. If the desired action is “make contact”, then the propensity to move towards the enemy will be increased for this state. In order to “wait for contact”, the movement speed can be set to zero and once one of the “in contact” states are triggered the agent will act accordingly. Adding the option of fortifying a position while waiting for enemy contact can be done by increasing the "number of hits to kill" value for an agent in the "next waypoint" and "in contact" states. The drawback to this is that it would apply to any "action in contact" during the simulation.

MANA has only the option for ending the simulation based on time steps. So validating this aspect of the framework through MANA will not be possible. It does provide a way to have multiple runs with a random seed variable that can be adjusted.

C. ARCHIMEDES MODEL DEVELOPMENT

1. General

The United States Marine Corps Combat Development Command's Project Albert sponsored Archimedes. Project Albert is focused on research into the behavior of complex systems and how it can assist military decision makers. Archimedes represents agents and the interactions between agents. The agents and interactions or connections are designed based on a template that is built. In this template, variables and aspects are added to the agents and connections to describe their characteristics.

The behavioral state (intent of the agent or connection) is represented by a collection of variables in a template. The variables are user defined and based of fuzzy logic. In fuzzy variables, variable values are divided into categories like "very close", "close", "near", "far", "very far" based on a max range or value. This allows agents to compare conditions, for example "if distance is less than near then set speed to fast."

The physical state (action part of the state) is based on aspects. Aspects are the means to represent things like movement, firing, and communication. These aspects translate information from the physical state to the behavioral state. An example of an aspect would be the *Movement Aspect;* based on movement strength, direction, and range, the aspect would update the position of the agent. There is also a "rules aspect" in which the interactions of the agents and connections can be specified using "if then, else" language and fuzzy logic rule evaluation.

The final part of Archimedes is creating a scenario. This entails designing a map in the map editor, assigning agents and the connections between the agents, setting the number of instances of agents that are to be generated and the agent's icons and colors. The simulation parameters can also be adjusted. This includes the number of time steps, the length of a time step and the random seed to be used. Once this is done the scenario is ready to be run.

2. Archimedes and Generalization

More detailed information about the use of Archimedes can be found in documents. Here we give only the generalization related types.

a. Combat Entities Generalization

In order to generalize combat entities in Archimedes, a basic infantry agent is created. Not related to the entity generalization, a number of variables need to be added to the basic agent. These include "injury status", "kill status", "receiving non-lethal fires", "training", and "speed". The added "speed" variable serves as our movement range characteristic. The durability can be represented by the "unit size" setting in the *State Aspect.* This setting is per agent, per instance and can be thought of as number of hits to kill. There is a communication aspect that determines if connected agents can communicate based on range and transmission probability. Since, agents are connected they can always be "sensed." In order to get around this, the rule aspect must be set so that actions will not take place unless agents are within a distance.

There is a weapons template where weapons can be built. So the entity characteristics of lethality, weapon range, and probability of hit adjusted on a template and then that template is added to the agents *Direct Fire Template.* This weapon template can address lethality in number of ways. A weapon can be set as lethal or non-lethal. If non-lethal is selected the weapons can have a variable probability to recover. A weapon's lethality can also be adjusted with the rate of fire variable and a variable to set if it is effective against soft and or hard targets. The weapon range can be set based on the minimum and maximum range variables. Finally, there is the ability to set the probability of hit for soft or hard targets at the minimum and the maximum range.

b. Combat Operation Generalization

The first of the operational points, the starting position, is set in an agent's *Position Aspect* where there is an "X position" and a "Y position" variable. Movement in Archimedes is based on one agent's connection to another agent and whether the "movement direction" is away or towards. In order to have an agent move towards a point, that point must be an agent. The exception is a waypoint agent. A waypoint is a collection of points that once one is reached the agents connection switches to the next waypoint and continues until the last waypoint is reached.

Action upon enemy contact must be defined in the rule aspect in the connection between the enemy and the friendly force. A unit can "hold" by setting movement strength to 0. It can "attack" by increasing movement strength. A "withdraw" action can be achieved by setting movement direction to away. A unit can push through if the movement strength towards the enemy is set to 0 and the movement strength towards another agent is greater than 0.

An effective way to manipulate the action at the objective point is to have an objective point agent. When a unit is within a set distance from it's objective, the *Rule Aspect* can be programmed with the correct action. The movement strength towards the objective can be decreased causing the connection a friendly force has with the enemy to take control, thus causing a "make contact" action. An agent's movement strength could be set to 0 so the rules in the connection to an enemy not to take effect until the enemy is within a set distance. This last group of setting can be used to simulate a "wait for contact" action.

Although there is no direct end of simulation settings other than by time steps, the connections between agents can be destroyed or reaped if a certain point is reached. In this case, the simulation will still run, but because there are no connections, nothing will happen. The reaching of this end mission point could also cause a 999 type of value to be placed in the data of the simulation run to indicate when this point is reached.

D. GENAGENT (GENERALIZATION AGENT) MODEL DEVELOPMENT

1. General

Beside the other two models (MANA and Archimedes) we developed our own MAS simulation model (GENAgent) to use to apply our generalization framework. GENAgent is an interactive, situated (including placement and coordinate dependent movement); high resolution, tactical level, MAS combat simulation. It is implemented in the JAVA programming language. Architecture and implementation depend on RELATE architecture, developed by Kim Roddy & Mike Dickson at NPS (Roddy and Dickson, 2000), and GI Agent by Joel Pawloski (Pawloski, 2001). GENAgent is developed to have the ability to simulate a wide variety of combat operations of different levels of tactical forces (from squad to company) by applying our generalization framework. Being a next generation and follow on work to GI Agent, GENAgent inherits the basic architectural and behavioral characteristics from GI Agent.

2. Architecture

GENAgent's architectural framework depends on GI Agent's implementation. It uses GI Agent's Line-of-Sight determination, path algorithm (A* search), its terrain and terrain features creation. For the details of line-of-sight, A* search algorithms, and terrain features refer to GI Agent (Pawloski, 2001). The RELATE dynamic relationship construction between the agents, has been applied to create different levels of force formations.

a. GENAgent Relationships

The Relationship construction between the agents in forces depends on a RELATE relationship as modified and applied in GI Agent. The agents of GENAgent are designed to have the ability to represent different types of entities (dismounted infantry, tank etc.) by utilizing the entity generalization principles. Agents can be organized in different sizes of tactical force levels-from single squad to a company. An agent's placements inside its unit and relationship creation principles are kept the same as in GI Agent. Although program has default values, the initial position of a unit in

GENAgent may be selected by the user and can be anywhere on the terrain map. This initial position is also recorded in operation ticket. Detailed information about the capability of placing units anywhere in the terrain will be given in next section. Once the unit is set in an assembly area, agents in the unit form their dynamic relationship depending on their environment and sensing range (See Figure 9).

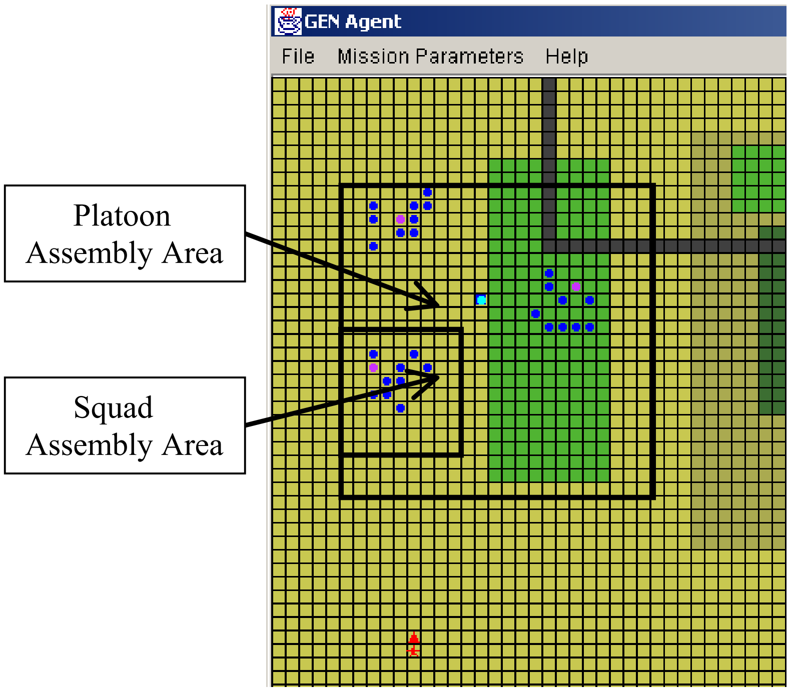


Figure 9   Placement of a Platoon on Selected Assembly Area

Since the RELATE paradigm needs agents to sense each other to establish a relationship, this capability can easily be used to create relationship of a core unit squad. This capability also restricts an agent's ability to establish another dynamic relationship with agents outside of its sensing range. This applies even if they are in the same, parent unit. So agents in different squads in a platoon would not be able to establish a "platoon relationship". GENAgent used GI Agents solution to the problem. The solution is to have a communication sensed environments besides the visual sensed environment. This enables agents to establish platoon or company relationship wherever the force is placed on the terrain regardless of the terrain features. This ability may be thought as a radio communication among the agents.

3. Capabilities

As mentioned above, GENAgent is a “follow-on work”, and in a way the next generation of GI Agent. In this regard it has inherited some architectural implementations and some of the common properties of GI Agent. GENAgent has improved upon GI Agent’s capability of simulating land combat in many ways by using the generalization framework. It has been designed to simulate a wider variety of combat operations with different force types and levels on any type of terrain. A user can create his own terrain, decide the size and type of forces to be used and define a variety of missions for the force to carry out. To help the dynamic relationship creation, GENAgent creates units in their assembly areas. Details about force placement are given in the following sections.

a. Terrain Creation

GENAgent’s terrain feature inherits the basic properties of GIAgent’s terrain building implementation. User can either select one of the pre-created terrain models from the menu or just create his/her terrain model. S/he has the ability to create a wide variety of terrain features (like water, elevation, road, cover) or modify any existing terrain map just by using GUI capabilities. The type of terrain on a map and where it is located can be changed by selecting the terrain feature from a GUI menu and then either clicking the mouse on a terrain square or selecting a group of squares where the terrain feature is to be place. Figure 10 shows a snapshot of the terrain creation feature. This capability enables simulation users to develop any terrain for a scenario. User can save the created terrain map as a file for later use.

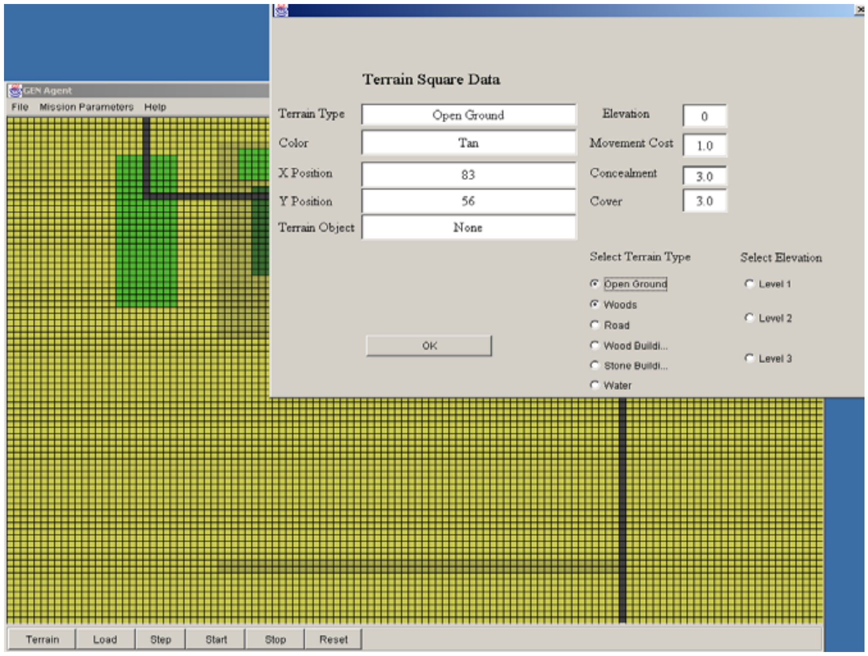


Figure 10 Creating/Modifying Terrain

b. Force Creations and Setup

Users can select the size of a tactical unit from squad up to company level in the Simulation Editor interface. The number of soldiers in this unit can be adjusted. This allows for the ability to create a squad, for example, with 10 agents in it or 15. GENAgent has the capability to create and handle a minimum 4 and a maximum 421 agents for one side. This maximum number is the upper bound and requires extensive computational power and CPU capacity. Figure 11 shows a portion of the interface for force setup. The user can also select up to 10 sniper agents for a defined unit. Once forces are created, organizational relationship among agents, or chain of command is established by RELATE's dynamic relationship process implemented in GI Agent.

Figure 11 Creating the Forces

c. Defining Mission Parameters (Operation Orders):

GENAgent has the ability to define any kind of operation in the format of the operation generalization framework. For executing an operation, a unit needs to be assigned an operation order that includes the details of operation. GENAgent has the capability to define a simple level operation order for each side. This includes the ability to define operation points (starting, objective, ending points and any waypoints), actions to be taken on these points, the maximum allowed casualty level before a unit quits an operation (breakpoint), and the waiting time limit for enemy contact at the objective (time out limits). As defined in the combat operation generalization principles, utilizing operation points and actions on these points can define any combat operation. A GENAgent user can define all of these attributes by just making selections on the GUI menu or by “clicking” the mouse on the selected terrain square. Figure 12 gives sample pictures of issuing operation orders for the units.

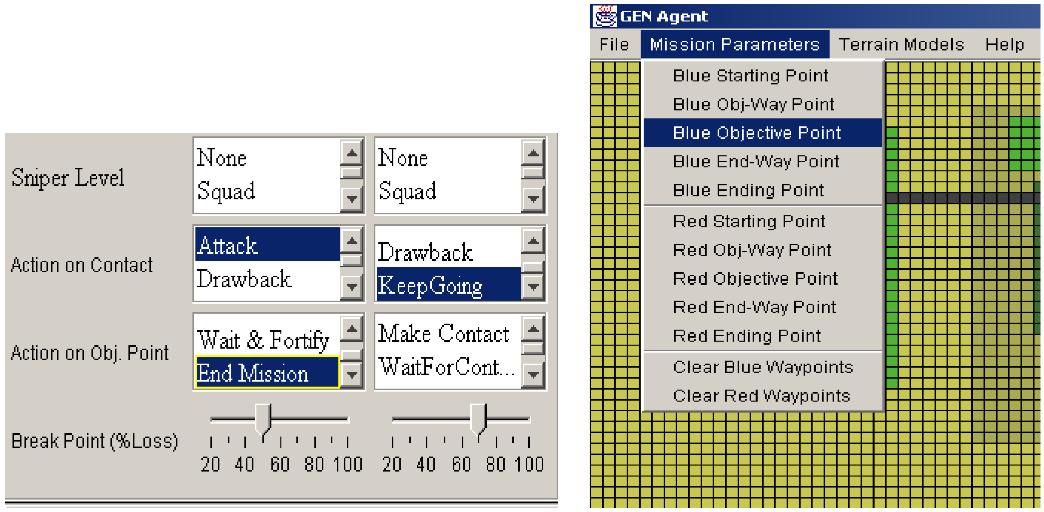
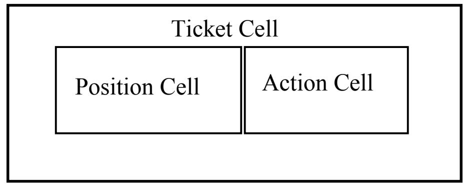


Figure 12 Mission Assignment

The operational order for each force, including the operation points and actions at these points, are kept in a special data structure called a "Ticket". (Hiles, 2001) The ticket is the issued order for each force. Tickets represent step-by-step procedural problem solving knowledge. Agents refer to their tickets when they need it for the operation execution. The issued information regarding operation execution is stored in "cells" of the ticket. The structure of a ticket and ticket cells are shown in the figure below. Ticket cells are loaded through the setting of simulation parameters. The "Action in Contact" selection is loaded to the "Starting Point", "End Point" and "Way Points" cells of the ticket. The "Action in Objective Point" selection is loaded in the "Objective Point" cell of the ticket. The position cells of the tickets are loaded after the selection of waypoints and operational points.

Starting P. Cell	Obj. Waypoints Cells	Objective P. Cell	End Waypoints Cells	Ending P. Cell

(1)



(2)

Figure 13 (1) Ticket Format (2) Ticket Cell Format

d. Force Placement:

Placement of the forces is also an important factor in combat simulations. Simulations should have a mechanism that allows for the placement of the forces anywhere at the beginning of simulation. In GENAgent, the forces can be placed anywhere on the terrain by just selecting the desired terrain square. The selected square will be a base point or the "starting point" for the force. The program will set up the assembly area and place the unit centered over this "starting point". The starting point is selected from a pop-up menu on the terrain window. User selects one of the operation points from pop-up menu and clicks on the desired terrain square to assign the point there. All other operational points and waypoints are defined in the same way. User can change any of these points by just reassigning the point to another terrain square and resetting the terrain. If a user-selected origin point is too close to the terrain borders to construct an assembly area, GENAgent shifts the origin in order to get minimum space requirements for an assembly area on terrain for the selected force level. A force can be placed anywhere on the terrain and it is up to the user to ensure that a force is not started over a lake.

e. Simulation Run Modes and Simulation Termination:

In GENAgent, a user can select one of two types of run modes. One is the single run mode. In this mode, a simulation runs until a force's mission is over. A mission is over when a casualty level has passed a breakpoint, a time out limit has been exceeded, or an endpoint is reached. When a mission is over the simulation stops and gives a warning message in a pop-up window that gives the reason the simulation is ending. In this window, the user can select either to resume, reset, reset & restart or stop the simulation. See figure 14 for sample mission over message.

The other option is data collection mode. In this option, the number of times the simulation will run can be selected. The possible number of runs in the data collection mode range from 20 to 100 times. In this mode, whenever one of the mission-over conditions is met, GENAgent restarts the simulation and writes the statistical output to a "dataout" file. Each run in this file is written on a new line separated by "tabs".

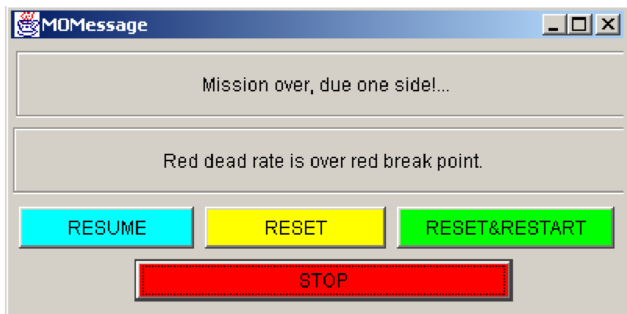


Figure 14 Sample mission over (MO) message

4. Design

The basic design architecture of GENAgent inherits some of GI Agent's design. The main java-based class that stars the simulation is AgentSim. This class initiates SimEditor. SimEditor provides the GUI for the simulation. It also is the starting menu and interfaces between SimEditor and AgentSimEnv, environment class that is the primary controlling class of the simulation execution.

AgentSimEnv mainly controls the three basic operations; terrain manager, agent manager, and mission manager. Terrain manager creates terrain squares and controls terrain operations. Agent manager creates blue and red forces and controls their operational behavior by utilizing interface classes. Mission manager controls the mission parameters. Mission manager also creates and holds tickets and controls mission executions. This basic structure is represented in figure 15.

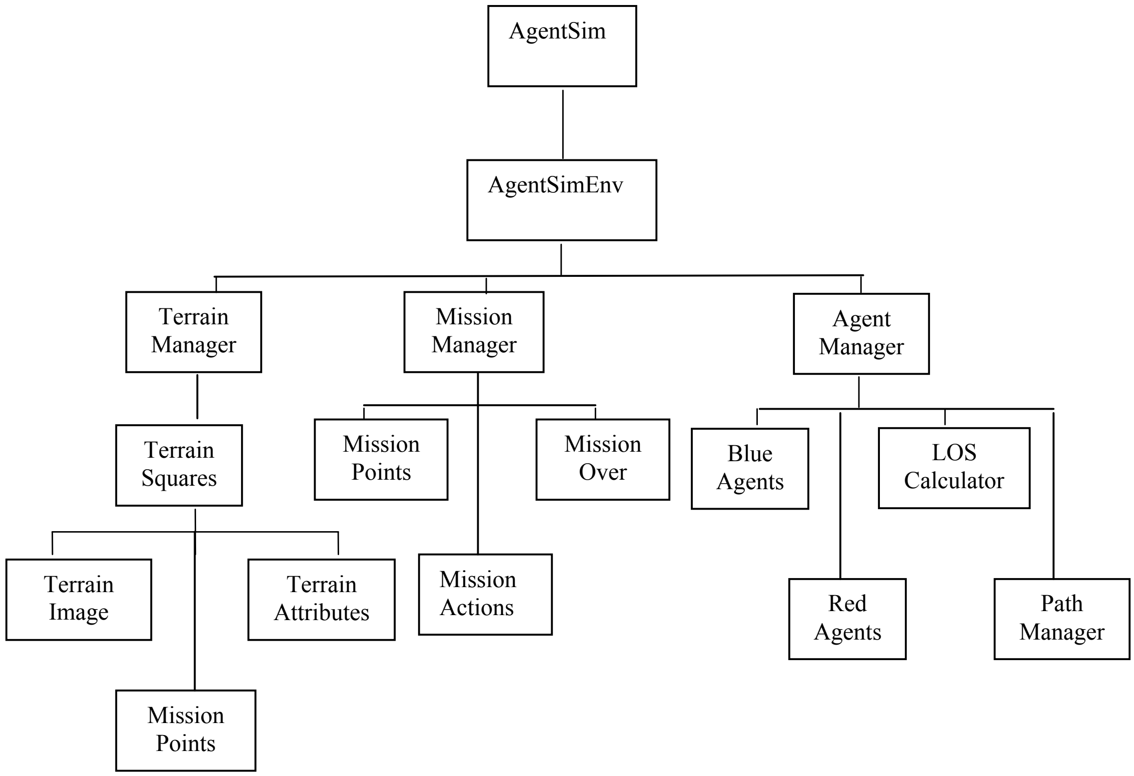


Figure 15 GENAgent Structural Design

a. GENAgent Simulation Editor:

The Simulation Editor GUI appears first when the simulation is run. This is an interface where many of the simulation parameters related to both agents and mission execution are set. The editor basically includes the components and slider bars for force setup, mission definitions, simulation run, and individual agent attributes for both blue and red forces. The figure 16 below shows the editor GUI.

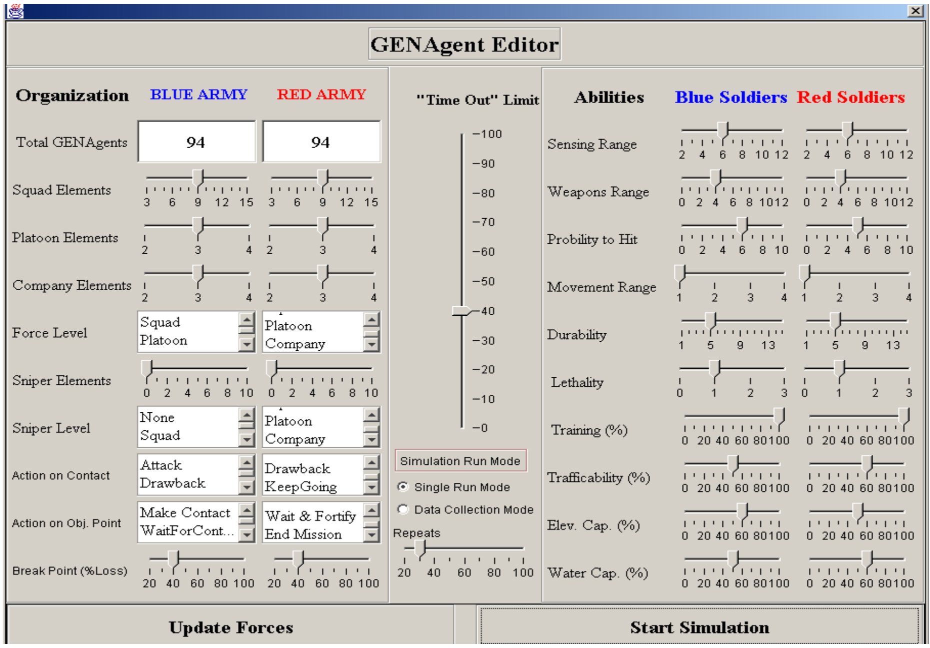


Figure 16 The Simulation Editor

On the left side of the GENAgent Editor the organizational properties and mission details can be set. The slider bars in the abilities section set the combat parameters for the agents in GENAgent. Explanation of each component is given below:

- **Total GENAgents**: Gives the calculated number of agents on each side after force setup is completed. It is updated whenever the **Update Forces** button is pushed.

- **Squad Elements**: Sets the number of soldiers or combat entities in a squad – ranging from 3 to 15. Default value is nine.

- **Platoon Elements**: Sets the number of squads in a platoon ranging, from 2 to 4. Default value is three.

- **Company Elements**: Sets the number of platoons in a company, ranging from 2 to 4. Default value is three.

- **Force Level**: Sets the force level or size of unit. If a "platoon" force level is selected the company slider value will be ineffective. Default force level is company.

- **Sniper Elements**: Sets the number of snipers to be added in a selected level. No sniper is assigned to forces by default. This is left over from GI Agent.

- **Sniper Level**: Defines the level of snipers to be added to the force. i.e. in a platoon level force with three squads, selecting sniper elements as one and sniper level as squad would result six snipers at platoon with two snipers at each squad. Default level is none.

- **Action on Contact**: Sets the action that the force will take upon any enemy contact on the way an operational point or waypoint. This is the action upon enemy contact during movement. This action is loaded to the ticket of the force. "Hold" option is not implemented in GENAgent currently. Default contact action is "Keep Going".

    - **Attack** - Unit attacks the enemy force whenever the enemy is detected.
    - **Drawback** - Unit withdraws to its base point - starting point.
    - **Keep Going** - Unit does not move towards the enemy, the unit keeps moving to its final goal. Unit can fire at the enemy as it continues to move as long as there is contact.

- **Action on Obj. Point**: Sets the action of the unit at the objective point. Default objective point action is "Make Contact".

    - **Make Contact** - Unit attacks the enemy upon reaching the objective point. If there is no enemy at the objective point, it searches for the enemy.
    - **Wait for Contact** - Unit waits for the enemy at the objective point.
    - **Wait and Fortify** - While waiting for the enemy, agents also get ready for the contact. They fortify the positions; increase their training and or durability.
    - **End Mission -** Whenever the unit reaches the objective point, it is assumed to have accomplished the mission. This option may be applied to a movement type of operation.

- **Break Point**: Sets the maximum allowable casualty level to give up the operation, in percentage of the total force. Whenever a casualty level of a force exceeds this limit, the simulation stops and gives the "Mission Over" message. Default break point is 0.4 for both sides.

- **"Time Out" Limit**: Sets the maximum time step limit for a force to wait for the enemy at the objective point. If there is no enemy contact during this time the simulation stops giving the "Mission Over" message. Default value is forty time steps.

- **Simulation Run Mode and Repeats**: **Run Mode** determines whether a simulation is going to be a single run or a data collection run. **Repeats** sets the number of times simulation will repeat in data collection mode. Default is single run mode

- **Sensing Range**: Sets the radius of visual sensing range for an ordinary combat entity. Sniper agents sensing range is set double that of a combat entity. Default sensing range is six.

- **Weapons Range**: Sets the range of a combat entity's weapon in squares. Sniper agent's weapon range is double that of the ordinary entity. Default range is four squares.

- **Probability to Hit**: Sets the probability a weapon shot will hit the agent it aims for. Default value of probability is 0.6 for ordinary combat entity and 0.9 for sniper agent.

- **Movement Range**: Sets the speed of an agent, or the number of squares an agent can move per turn. Default value is one square. Due to time constraint this selection is not implemented, so it is not active in the simulation execution. It is left for future work.

- **Durability**: Sets the maximum health count, which is the number of times an agent can get shot before it dies. Default value is five.

- **Lethality**: Sets the lethality level of agents. This is actually the lethality of weapon an agent is using. This is the amount of decrease in durability level when an agent hits another agent. Default value is one for an agent and two for sniper. Lethality range is from 0 to 3.

- **Training**: Sets the level of training or combat readiness of a force. Training level affects the probability of hit. It can be improved and extended for other options as well. Default value is 100%

- **Trafficability**: Sets the movement capability of agents in percentage on terrain. Gives different movement capabilities to different types of entities through entity generalization. Due to time constraint this selection is not implemented, so it is not active in the simulation execution. It is left for future work.

- **Elevation Capability**: Sets the agents capability in percentage to navigate through terrain elevation. Used for entity generalization. Due to time constraint this selection is not implemented, so it is not active in the simulation execution. It is left for future work.

- **Water Capability**: Sets the water traversing capability of agents. Due to time constraint this selection is not implemented, so it is not active in the simulation execution. It is left for future work.

- **Update Forces**: Calculates the number of agents in a selected level of force after selections have been made in the organizational slider bars. It also updates the selected options.

- **Start Simulation**: Sets the all chosen parameters to an environment class and starts the simulation by bringing up the terrain window up.

b. Terrain and Terrain Manager:

The terrain manager embedded in AgentSimEnv handles terrain operations. The terrain is composed of square objects holding all the necessary terrain information. It holds information about the objects terrain feature and any agent presence. Detailed information about terrain squares, terrain operations and terrain management can be found in GI Agent (Pawloski, 2001). Terrain objects in GENAgent have been modified to contain information if they are operation points or waypoints and holding the mission points as symbols on them. Any terrain square assigned to be an operation point or waypoint still holds its basic terrain and agent information. A square containing one of these points is just a symbolic reference for agents to count. Terrain features can be viewed and or adjusted by selecting a single terrain square or a block of squares. When a square is selected a Terrain Square Dialog Box pops up and user can view and change the feature of that square. To access a terrain block, user clicks and drags the mouse to include a group of terrain squares. This action pops up the Terrain Block Dialog Box.

c. Agents and Agent Manager:

GENAgent agents are created and controlled by an agent manager. It is embedded in the AgentSimEnv class and is the master application that controls agent operations. Agent manager enables agents to interact with the terrain and get the parameters from the user interface. At the beginning of the simulation, it creates the agents around the selected starting position in the assembly area. Agent manager also controls the line-of-sight and path calculations with the LOSCalculator and PathManager classes, respectively. Line-of-sight and path calculations, managed by these classes, at each step of the simulation are the major overload on the execution. During each step of a simulation these calculations are done for every agent. If there is large number of agents in a simulation, these calculations can become an overload on the computer.

The basic class that holds the agent structure is GENAgent. It has all the parameters of an agent and acts as the agent object in simulation. Detailed information about agent manager, Line-of-Sight Calculation and Path Manager can be found in GI Agent (Pawloski, 2001).

d. Mission Manager:

The mission manager controls all mission related activities. It is implemented in the MissionManager class and is controlled and utilized by the AgentSimEnv class. MissionManager fills up and assigns tickets for the forces, handles the mission points over the terrain, controls the execution of active ticket operations, checks for enemy contact, manages objective shifts while navigating through the terrain and checks for end of mission conditions.

Tickets are controlled in mission manager with functions for ticket creation and ticket cell assignment. Once the tickets are created and loaded, MissionManager sets the initial ticket cell as an active ticket cell. Throughout the simulation run, once a mission point is accomplished, MissionManager shifts the active ticket cell and sets the next ticket cell as active for agents to refer to.

MissionManager also keeps track of the mission actions to execute. These actions are kept inside the action cells of the ticket. These action options tell the agents what to do upon enemy contact. For example, if the action on contact is “Attack” and the force has enemy contact, MissionManager shifts the agent’s objective point temporarily to the enemy position until the enemy is killed or contact is lost. Then it resumes the active ticket cell.

Another important job managed by MissionManager is to check for mission-over conditions in every step. If one of the mission-over conditions is met, the manager stops the simulation and gives the user the “Mission Over” message or restarts the simulation, if multiple run mode was selected.

e. Multiple Terrain Options:

The user can create the terrain in GENAgent or a previously created terrain can be opened. Terrain models are opened from the "Terrain Models" pop-up menu in terrain window. The menu has four options, a plain terrain and three different previously designed terrains. Figure 17 below shows the menu selection.

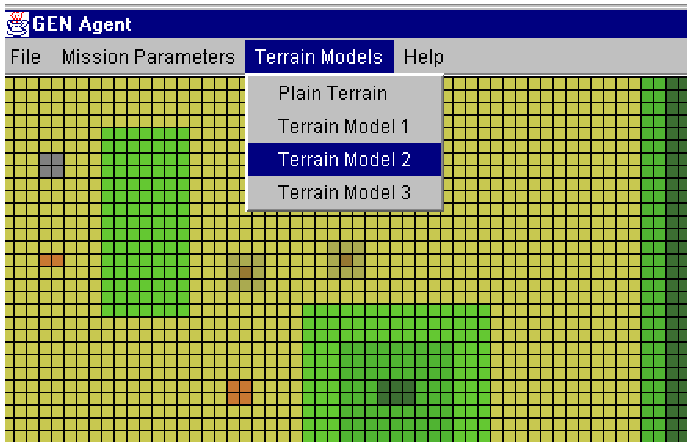


Figure 17 Selecting Terrain Models

5. GENAgent and Generalization

GENAgent has been specially developed to utilize and test the generalization framework. So the settings are designed to meet the requirements of generalization framework. Any setting can be changed or reset anytime through the simulation execution by selecting the "Back" option in "File" pop-up menu in the terrain window. The new setting does not become active until the simulation is reset.

a. Combat Entities Generalization

The simulation editor contains the interface for setting basic agent capabilities in the "abilities" section. The settings are applied to all the agents under the indicated force. An agent's visual sensing range and weapons range can be set through slider bars. There is also an option to set the probability of hit for a given agent. This determines the ability of an agent or weapon to hit enemy agents and is affected by the training level of an agent. Movement range is set to a constant value, and for time constraints has not been implemented. All agents have the same movement capability. Slider bars are used also set durability and training level of an agent. Trafficability, elevation capability, and water capability options are also left for future work since time constraints.

b. Combat Operations Generalization:

Force setup, force levels and number of different elements in the force, are set at the beginning of the simulation. The three main operational points and the waypoints between them can be set through the pop up menu in the terrain window. The end-waypoints, waypoints between the objective point and the ending point, have not been implemented. These points would have the same functionality as the objective-waypoints or waypoints between the starting point and objective point. The starting point is where a force will be placed initially; objective point is where the main mission is executed; and ending point is where the force goes at the end of the mission. If no waypoints have been defined between the starting and objective points, agents will go directly to the objective point following the optimum path calculated by the A* algorithm that uses terrain elevation changes, cover and concealment, enemy and friendly forces, movement cost of terrain and agent's internal factors as heuristic function parameters. More detailed information on the A* search algorithm for path finding may be found in GI Agent (Pawloski, 2001). But users can define another path between starting and objective points by defining one or more waypoints. The actions to be taken on enemy contact and at the objective point are also set in the simulation editor. Breakpoints for the forces can also be set ranging from 0.2 to 1.0.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. SCENARIOS AND EXPERIMENTS

A. CHAPTER OVERVIEW

The three models discussed in the previous chapter were used to simulate two different combat scenarios. These scenarios were based loosely on scenarios found in the Marine Corps Gazette's "Mastering Tactics – A Tactical Decision Games Workbook" (Schmitt, 1993). The term loosely is used because these scenarios contain a high level of detail, and we are modelling them with distillation modelling tools. The first scenario is entitled "Ambush at Dusk" and involves an ambush on dismounted infantry. The other scenario is entitled "Enemy over the Bridge" and involves a mechanized assault on a defensive position. These two scenarios were chosen because they include a wide range of different entities and operations. The analysis is not comprehensive or complete, just a representative to demonstrate the generalization framework.

B. GENERALIZING THE SCENARIOS

1. Ambush at Dusk

In this scenario the friendly force or "blue" force is conducting a security patrol and is ordered to attack and destroy any enemy force discovered. The friendly unit is a squad reinforced with a machinegun squad, which includes 2 machinegun teams. The patrol is moving north through a rice paddy with a village to the west. While passing the village, automatic weapons open fire from the village. The size of the enemy in the village is unknown. Intelligence reports indicate that enemy guerrilla forces in this area are armed with small arms and light machineguns.

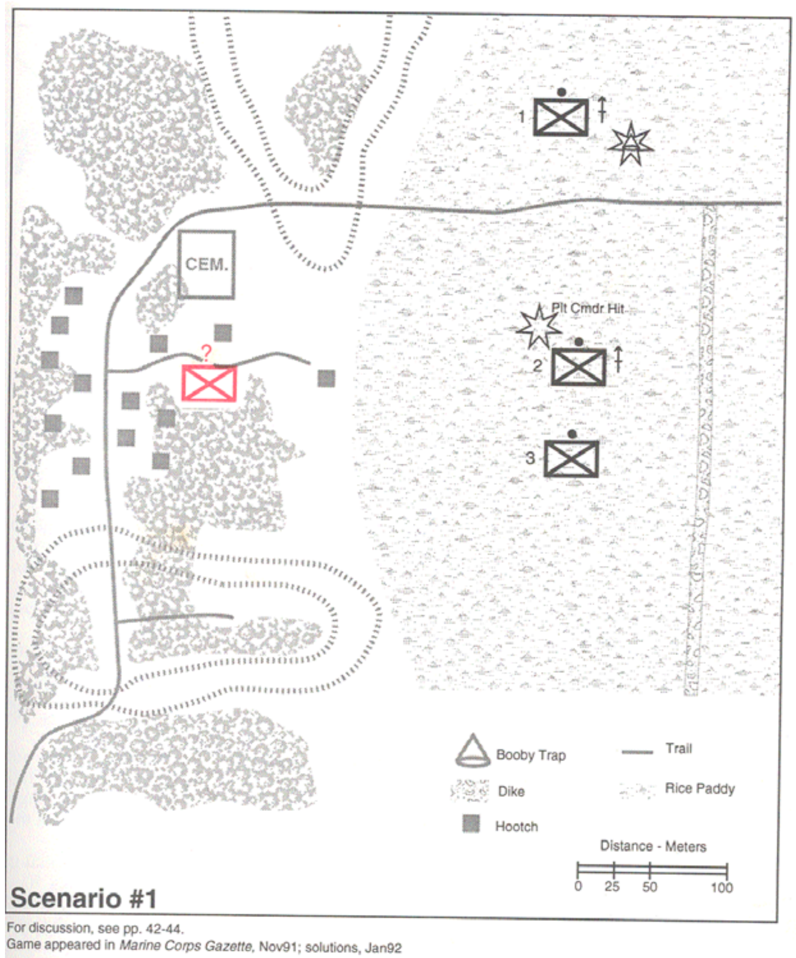


Figure 18 Scenario # 1 Ambush at Dusk

This scenario needs to be translated into generalization form. For the blue force, the entity characteristics will include straightforward ratio settings for a squad of infantry and two entities where the lethality is increased to represent the machinegun team. The red force is of unknown size and strength. Because the enemy is a guerrilla force it can be presumed that it is a small unit and equipped with the small arms and light machineguns.

The patrol can be thought of as a type of movement operation, where the starting point will be at one end of the area of operation and the objective point and endpoint will be at the other end and co-located. For this movement operation, arriving at the final waypoint/objective point/endpoint completes the operation. This arrival can be set in a type of end mission criteria. The action in contact will be “attack” based on the instructions to attack and destroy the enemy. The red force is set in an ambush in the village, so the starting point, objective point and endpoint will all be the same point. They start in the village and attack once the blue force is in the right position. Their action in contact will be to hold. The action in objective point will be to “wait for contact”. Since they are a guerilla force their start and objective points are in the village and their endpoint is some withdrawal point in the jungle when they reach some designated breakpoint.

2. Enemy Over the Bridge

This scenario involves a larger battalion size force. The battalion consists of two rifle companies reinforced with Dragons and heavy machineguns on foot, one company on trucks, a weapons company, a tank company, and a TOW section on HMMWVs. The situation involves a bridge that is occupied by enemy forces with light vehicles and another infantry enemy force south of the bridge, west of the town of Hamlet occupying blue’s proposed assembly area. The mission of the blue force is to prepare the area for a regimental attack in which this blue force will be the spearhead of the operation.

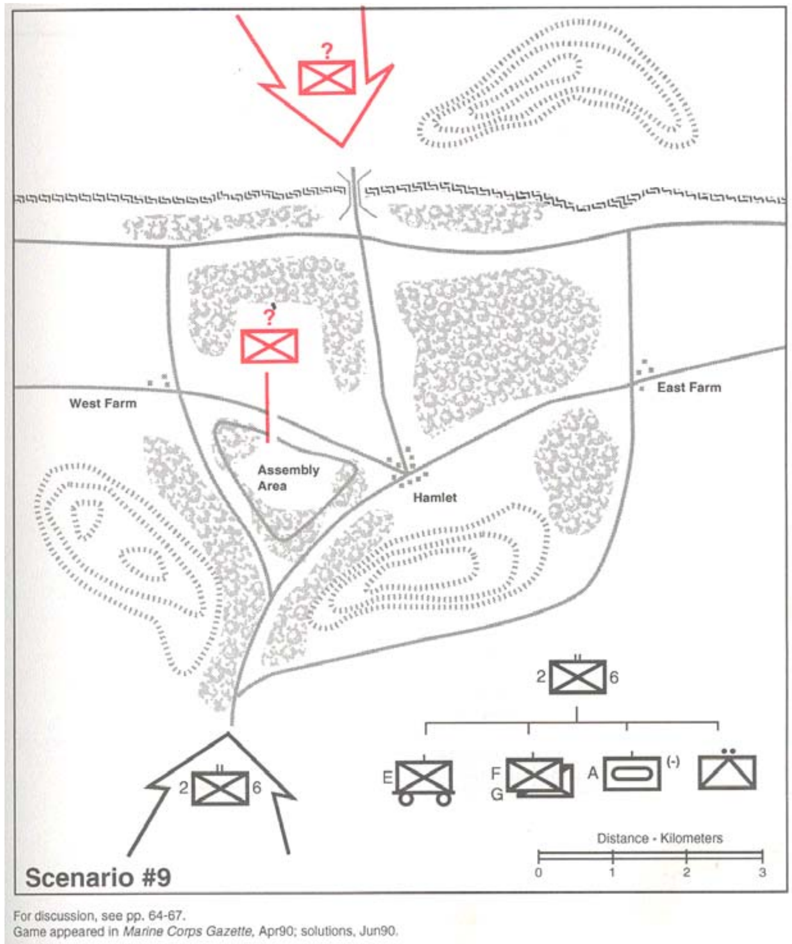


Figure 19 Scenario #2 Enemy Over the Bridge

The entities for the blue force in this situation are very diverse. All of the characteristics for the entities need to be adjusted differently to represent the different capabilities of these units. Among some of the capabilities that need to be addressed are the tank's high durability, the TOW section's high probability to hit and lethality, the trucks fast movement range, and the Dragon's high lethality. The red force is not a diverse, but still has two different types of units which are what and where are they located.

The operation will involve the splitting up the battalion to attack the two enemy positions. Both of these attacks will be of the assault nature, with possibly the foot mobile force attacking the closer enemy force and the mobile force using its speed to attack the enemy at the bridge. Both of the blue forces start in the same location, with the objective points being at their respective enemy unit, and the endpoint being the bridge. The red forces are in a defensive posture with all three points the same, with wait and fortify, and attack on enemy contact.

3. Scenarios built in MANA

a. Ambush at Dusk

For this scenario, a different MANA "squad" was designed for the basic riflemen in the fire teams and the machinegun sections. The fire teams have basic infantry values for their characteristics, while the machinegun teams have a larger "firing range" and "max targets per step". The red force is made up of entities with basic infantry values, with slightly less durability.

Figure 20 Ambush at Dusk in MANA

The red agents have all three operational points the same. This guerilla force, starts in the village and attacks from the village and has no plans for retreat. The red force is "waiting for contact" and has no movement when there is no enemy contact. Upon enemy contact there is no movement towards the enemy and just a value of 1 for movement speed.

The blue force utilizes the one-waypoint method to represent its objective point and endpoint. It does have another waypoint, but this is a true waypoint and serves to get the blue force closer to the village. Upon enemy contact blue is set to attack, so its propensity is to move towards the enemy. There is no action at the Objective Point. The blue force's end mission criterion is reaching this point.

b. Enemy Over the Bridge

This scenario is much more complicated than Ambush at Dusk. To avoid overloading MANA, aggregation of forces was used. The red side is aggregated into two forces and the blue force is aggregated into 6 different units. The six different units are the two dismounted companies, the tank company, the anti-tank section, the truck mounted infantry company, and the headquarter company.

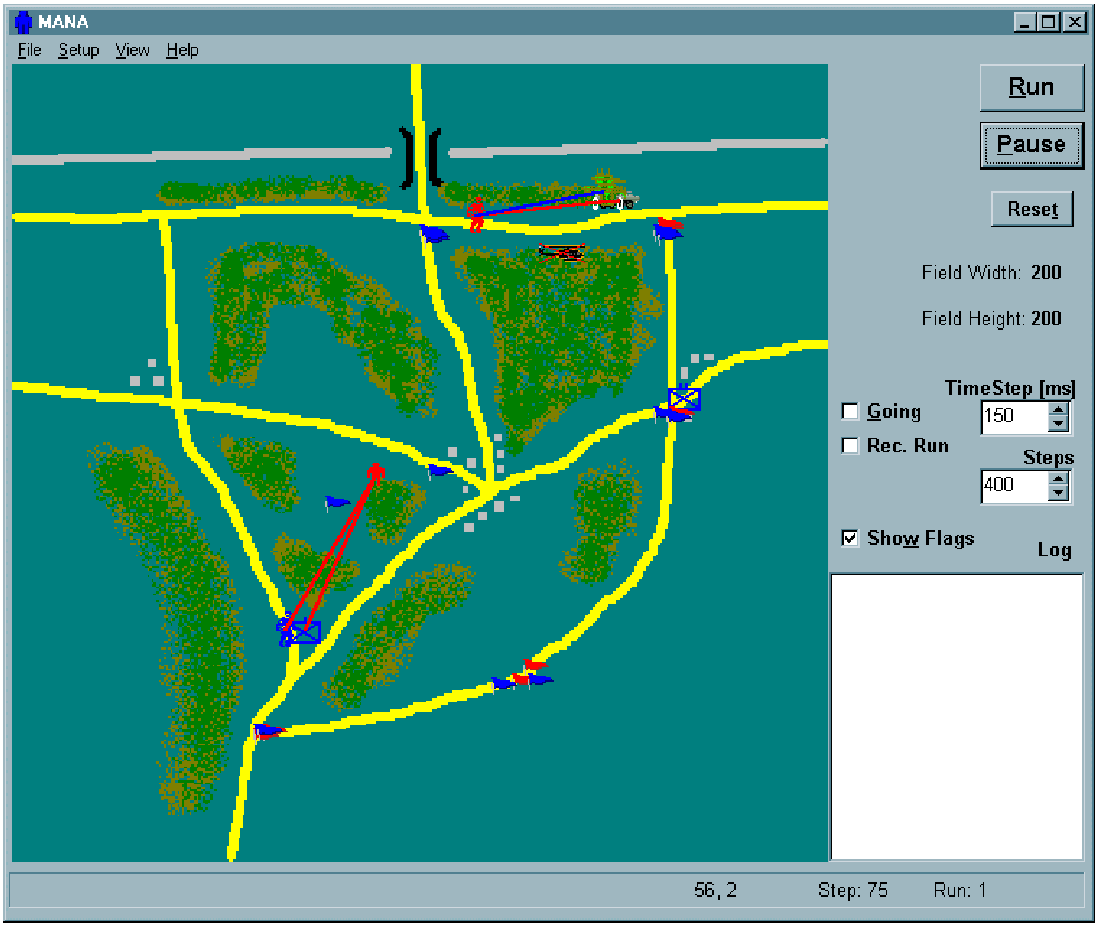


Figure 21 Enemy Over the Bridge in MANA

The characteristics for the red forces are set to a normal infantry ratio. The only difference is that the enemy at the bridge has a higher number of hits to kill and a larger number of targets that can be engaged at one time. This is to represent a larger unit than the unit in the assembly area. Both red forces are placed in their respective positions and have all three operational points the same. They are essentially in a defensive position. Upon enemy contact they will “hold” their position and at the objective point (where they start) they will “wait for contact”.

The blue force is divided into two units. They all have the same starting point at the bottom of the map and ending point at the bridge. The first unit is composed of the two foot-mobile companies. They are given basic infantry characteristics. Basic infantry characteristics can be described as values for the generalization characteristics of a combat entity set to reasonable values. These reasonable values would be similar to a 5 mile per hour movement speed, a 500 yard maximum effective weapon range, a weapon with a low lethality rate, a sensing range of 1 mile and communication range of two miles, a probability of hit around .75 percent and the lowest durability value (or next to lowest if civilians were to be modeled). Upon enemy contact they are ordered to attack. Their objective is the assembly area and is to make contact upon reaching this point. The other unit is a highly mobile force with all forces being vehicle mounted and one tank company. The characteristics of the HQ Company and the truck company are set to basic infantry values, with a higher speed due to their being transported by the trucks attached to the battalion. The speed of the HQ company is set one less than the truck company in order to represent the trucks coming back for the HQ company after they drop their first load. The tank company's values are set so that it has a greater weapon range, number of hits to kill, and firepower or lethality value. The anti-tank section's weapon range and its lethality are the same as the tank companies, but it has a number-of-hits-to-kill value larger than an infantry's but significantly less than a tank's values. The speed of the anti-tank section and the tank company is set one unit larger than the truck company to get them to the front of the formation. This mobile force uses a series of waypoints to get to the bridge by a round about way to avoid the assembly area. They are programmed to attack upon enemy contact. The objective point is just south of the bridge and they are programmed to make contact with the enemy there.

4. Scenarios Built in Archimedes

a. Ambush at Dusk

As in MANA, some aggregation was used in the scenario development in Archimedes so as not to slow down the time steps. Three copies of the basic infantry agent were made. One is used as the red force and the other two are the blue force. Two agents are used for the blue force in order to model both the infantry fire teams and the machinegun sections. The red force and blue non-machinegun agent's values were left the same. The machinegun agent's values were also left the same, but a new weapon template was built. This machinegun weapon template has an increased rate of fire over the M16 template that; the other two agents' possess. This increase in the rate of fire represents a greater lethality for the machinegun.

The red force was not provided with a position agent. It is initially placed in its ambush position and has connections with only the blue force. Upon enemy contact it "holds" its position by keeping a 0 movement strength, because it is not moving this also creates a "wait for enemy" at the objective. The red force does not start firing until the blue force is within a "near" range. The blue force does not start firing until after the red fires first, thus representing an ambush. The blue force starts at the bottom of the map and works its way north. Upon enemy contact, which in not until red opens fire, blue "attacks". The objective point for blue is the top of the map. It is conducting a movement or patrol operation and once it reaches it's objective point it is at the end of its mission.

b. Enemy Over the Bridge

For this simulation four copies of the basic infantry agent are made. There are two copies for each of the red and blue forces. This is more aggregated than what was done in MANA, but having eight different agents in Archimedes would result in having to define close to twenty or more different connections. This many agents and connections would slow the performance of the modeling tool.

Both red forces have basic infantry characteristics and only the durability is adjusted to reflect a larger red force by the bridge. These red forces are in a defensive position; they are placed in their respective spots and remain there until enemy contact is made. "Upon enemy contact" both red forces slowly increase their movement strength towards the enemy. Red is at its objective point and "waits for contact", simulated by having 0 movement strength until blue is within a certain distance.

The two blue forces start the simulation at the bottom of the map and have an ending point just south of the bridge. One blue force attacks the enemy at the assembly position. This blue force represents the foot mobile infantry company. So the characteristics for this force are that of the basic infantry agent. The foot mobile force "attacks" the enemy upon contact. At its objective point, the assembly area, the blue force will "make contact" with the enemy. The other blue force represents the mechanized part of the battalion. This agent has an increased durability, lethality, weapon range, speed, and probability of hit. It has an objective point of just south of the bridge, but it is going to get there by way of a series of waypoints that represent it taking the long way around the map to avoid the assembly area and flank the red force at the bridge.

5. Scenarios Built in GENAgent:

a. Ambush at Dusk:

For the ambush scenario, the terrain was created in 90 degrees rotation because of the landscape window properties. The blue force has been created as a platoon with three squads. Their mission was to start from the left end of the terrain and follow a defined patrol path to the objective point on the other end. The patrol path was set with one waypoint that brought them closer to the red force. The ending point is at the same location as the objective point since the mission is patrolling or a movement operation. The blue's contact action was set to attack, and objective point action was set to end mission since their mission was over when they reached this point. The red force was created as a unit of ten agents. Their mission was to set an ambush to the blue. The starting and ending points were set to same place and objective point was set close to the expected blue path. Their mission was to start at the starting point, lay an ambush at the objective point and after the operation come back to the end point. The red's action on contact was set to draw back and objective point action was set to wait for contact.

b. Enemy Over the Bridge:

This scenario was more complicated from Ambush at Dusk, and needs more than one force to be created for each side. Since GENAgent in its present version has the capability to create and handle only one homogeneous force for each side, and this scenario involves a blue force with mobile and dismounted forces and a red force with two units in different locations the simulation is divided into two different parts. One simulation was used to model the interaction of blue's foot-mobile force and red's infantry forces in the assembly area south of the bridge. Another simulation modeled blue's highly mobile forces and red's units at the bridge. Since GENAgent is capable of creating and handling at most company level unit size we aggregated the units into one company by adjusting agents capabilities appropriately. In both scenarios blue forces had the same starting point and objective point. But waypoints were set differently to make the paths different. Both of their contact actions were set to attack and objective point actions were set to make contact. In the scenario involving blue's dismounted forces, all three points of red forces were set to the assembly point at the south of the bridge. In the other scenario, red's starting, objective and ending points were set at the bridge. In both scenarios red's action on contact is set to attack and action in obj. point is set to wait and fortify. Again, because of the map window properties, map was created in 90 degrees rotation.

C. STATISTICAL EXPERIMENT WITH GENAGENT

Our aim for this analysis is to show that we can gather useful data generated by the simulations, not to prove or disprove military theory. Our purpose is not to get scientific statistical conclusions with statistical analysis on the simulations. Archimedes and MANA are existing simulations and we will not attempt to prove that their data generation is valid because it is out of the scope of this thesis. Rather, we use them to support our thesis that the generalization framework is applicable and effective to any high-resolution agent-based combat simulations. For the statistical analysis part, we will analyze separately the data taken from GENAgent's runs from the two scenarios. The analysis is not a comparison of the two scenarios to each other or a comparison of the models.

1. Scenario One: Ambush at Dusk:

In this ambush simulation the blue force level was selected as a platoon with 3 squads and the red force level was selected as a squad with 9 elements. With this scenario, we analyzed the outputs of three different settings of red's lethality in the same conditions, holding all other parameters the same. In the different simulation runs, we set the lethality of red forces to 1, 2 and 3. This represented the red entities' different capabilities in combat power. The simulation was run 30 times for each setting to collect data. Simulation time is the time for a simulation to reach an end mission point. This represents a defeated enemy or a force reaching its endpoint. We expected to have different reasonable results for each setting reflecting the changes in red's lethality level. Appendix C shows the exact parameter settings of forces for this scenario.

At the end of each 30 repeats with one setting, GENAgent stored results in a "dataout" file. The analyses of these three different data sets in Microsoft Excel gave us the results. Increasing the red force's lethality level at each time caused an increase in simulation run time for mission over (operation time), increase in blue death rate and decrease in red death rate respectively.

The first set of data results with red lethality level of 1 is listed below.

The average simulation time was 69.10 time steps with a standard deviation of 2.70.

The average blue death rate was 12.68 % with a standard deviation of 5.73.

The average red death rate was 60.33 % with a standard deviation of 1.82.

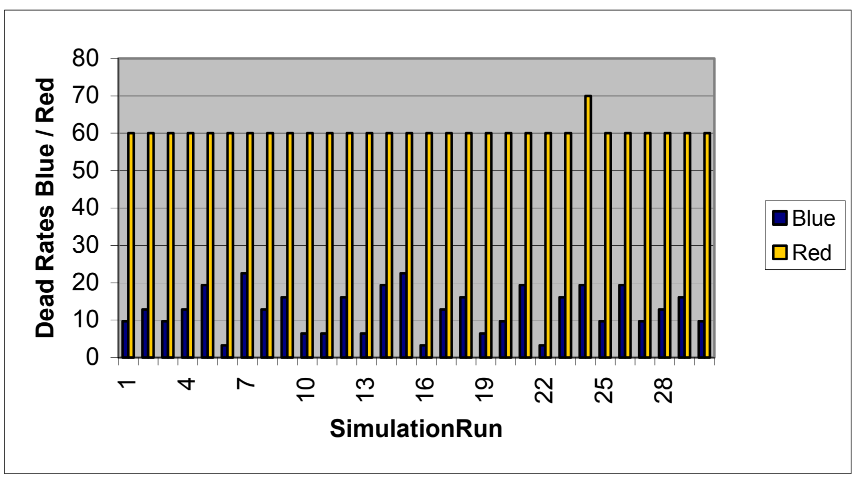


Figure 22 Blue and Red Death Rates for Red Lethality = 1

The second set of data results with red lethality level of 2 is listed below.

The average simulation time was 74.90 time steps with a standard deviation of 2.66.

The average blue death rate was 27.20 % with a standard deviation of 9.72.

The average red death rate was 59.33 % with a standard deviation of 4.49.

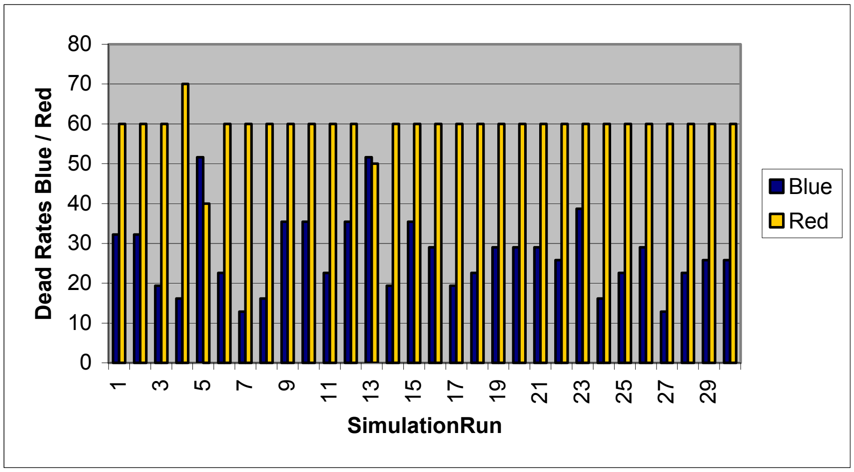


Figure 23 Blue and Red Death Rates for Red Lethality = 2

The third set of data results with red lethality level of 3 is listed below.

The average simulation time was 75.10 time steps with a standard deviation of 1.55.

The average blue death rate was 46.98 % with a standard deviation of 7.60.

The average red death rate was 40.33 % with a standard deviation of 17.11.

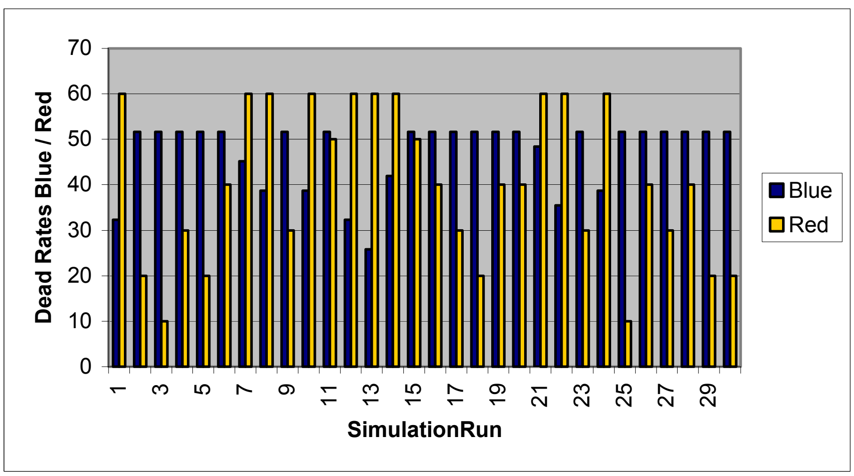


Figure 24 Blue and Red Death Rates for Red Lethality = 3

From the analyses results we can see that as the red force's lethality level increases, blue's casualty level and the simulation time increase and red casualty level decreases. Reasonably, when red combat power increases, it takes more time for blue to overcome the threat, and causes more casualties. There is a slight increase in simulation time when red lethality level is increased from two to three. When red's lethality is set to three, blue does not always win the combat action, and the time does not represent only average time for blue to defeat red. At this level, each side takes longer to get to the other's casualty level to a breakpoint. Figure 25 shows respective casualty levels of blue and red for each setting.

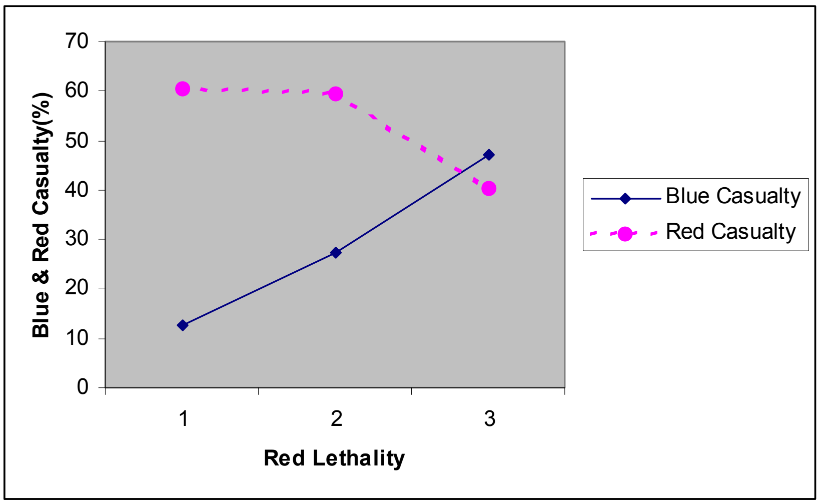


Figure 25 Blue and Red Casualties vs. Red Lethality

2. Scenario Two: Enemy Over the Bridge:

In this scenario we split the simulation into two different parts. Each part was a separate simulation run with different setups and different data outputs. The first part was that blue's two rifle companies against red's one company in the assembly area. Since GENAgent is capable of handling at most company-level forces, blues' two companies were aggregated into platoons. The simulation was thus set up with two platoons to represent blue's two companies and red with one platoon. Appendix D shows the force parameters of both sides. In this scenario, we analyzed the results of three different settings for the training level for the blue forces. The training levels were set to

100%, 80% and 60% respectively. We expected to see that blue's casualty level and simulation time increases as blue training level decreases.

Again, Microsoft Excel was used to analyze the data from the 30 repeat runs for each of the three data sets. Decreasing the blue's training level each time caused an increase in simulation run time for mission over, which is operation time, an increase in blue death rate and a decrease in red death rate.

The first set of data results with blue training level of 60% is listed below.

The average simulation time was 59.40 time steps with a standard deviation of 2.98.

The average blue death rate was 35.55 % with a standard deviation of 5.90.

The average red death rate was 70.97 % with a standard deviation of 0.

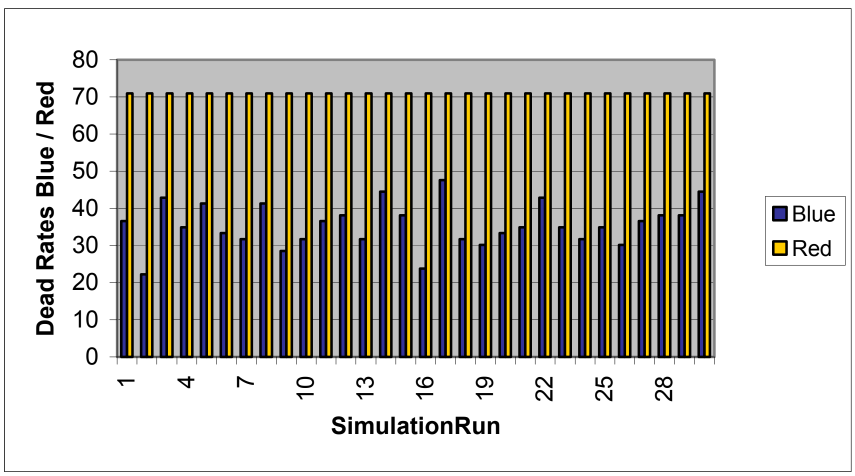


Figure 26 Blue and Red Death Rates for Blue Training = 100%

The second set of data results with blue training level of 80% is listed below.

The average simulation time was 61.20 time steps with a standard deviation of 2.78.

The average blue death rate was 42.16 % with a standard deviation of 7.54.

The average red death rate was 70.97 % with a standard deviation of 0.

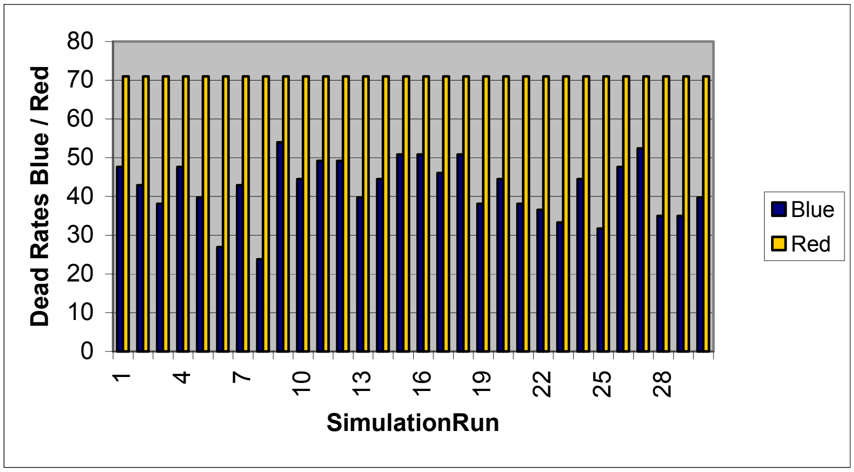


Figure 27 Blue and Red Death Rates for Blue Training = 80%

The third set of data results with blue training level of 60% is listed below.

The average simulation time was 67.70 time steps with a standard deviation of 3.24.

The average blue death rate was 54.76 % with a standard deviation of 5.71.

The average red death rate was 66.45 % with a standard deviation of 8.06.

With the first two settings above the red force loses every time. With a breakpoint of 70% the simulation restarts because red force's casualty level exceeds that level before blue each time. So the standard deviation is 0.

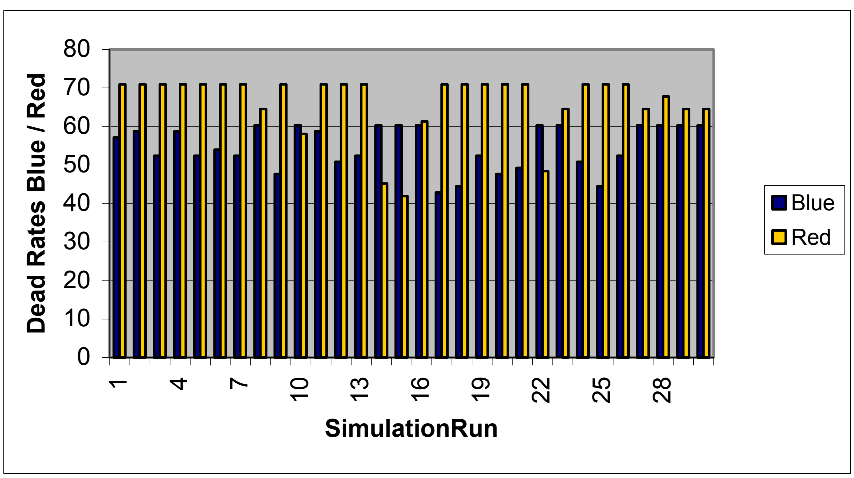


Figure 28 Blue and Red Death Rates for Blue Training = 60%

The experiment results show that, as blue training level decreases, the blue casualty level, the simulation time, the time for blue to defeat red, increases. Conversely, red's casualty level decreases. These results justify our expectations. Figure 29 shows respective casualty levels of blue and red for each setting.

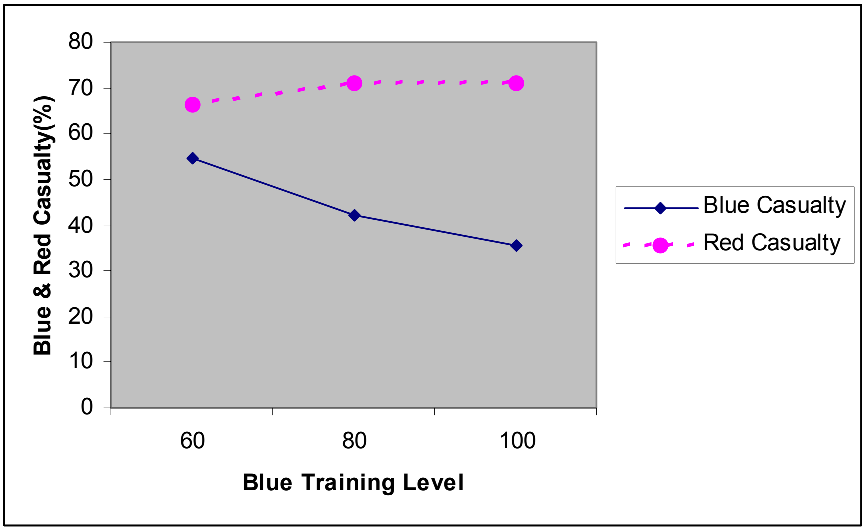


Figure 29 Blue and Red Casualties vs. Blue Training Level

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSION

A. RESULTS

The two scenarios, ”Ambush at Dusk” and “Enemy Over the Bridge” were very detailed tactical decision games. All the details of the games are necessary in order to assist a unit leader in his decision making process for a course of action for his unit. The details can get in the way of some of the aspect that an analyst may be trying to examine. The framework allowed the situations and the units involved to be broken down to their basic form to be better suited for use in a MAS.

In MANA and Archimedes the process of building the simulations for the scenarios showed how an analyst would benefit from the framework. MANA took the shortest time to set up. The framework for the combat operations and combat units was easily transformed to the trigger states and variables in MANA. The help file in MANA allowed for easy understanding of its capabilities. MANA had the easiest to use and straightforward GUI interface of the two existing simulations. MANA did limit the user to the variables and parameter ranges already present. Archimedes did not have this limitation and allows the user to create as complex a simulation as desired. This ability also contributed to the longer amount of time that was needed to understand Archimedes. Archimedes also did not have a detailed help file, which contributed to the steep learning curve. The framework was also not as easily correlated into the aspects and connections in Archimedes as in MANA. The framework could still be applied to it, but it just lengthened that amount of time that was needed to build the first scenario. A table comparing the modeling tools is included in Appendix E.

GENAgent was created based on the framework and illustrated the benefits of the framework to a developer. The design of GENAgent took a long time, but once the simulation was built, modeling a new scenario was relatively quick and simple process.

1. GENAgent Experiments

As expressed before, we did the experimentation by running of the two scenarios in GENAgent. We did not go into the validation of experiment results. The purpose was just to see whether we could get reliable and reasonable results with the data from the simulation runs with different settings. And after the analyses of the two different sets of data, we could see that varied settings have given reasonable different results.

In the first scenario, Ambush at Dusk, the red lethality level increases in a consequent set of runs. The analyses results show that the average simulation time and average blue and red casualty levels are affected reflecting the changes made. The increase in average simulation run time reflects the time increase for blue to overcome more powerful enemy force at each time. And the corresponding differences in average casualty levels also show realistic effects of changes made in simulation setting. In the second scenario, Enemy over the Bridge, the analysis results reflect the effects of different blue training levels over simulation results. These results, as explained above, again reflect the reasonable effects of different settings.

Overall, the results show us we can get dependable and realistic data from the execution of the simulation. In the two experiments with GENAgent no non-obvious results were observed. If more variables were adjusted between simulation runs, non-obvious results might have been observed. It shows that you can gain insight into the problem domain. The results also showed that GENAgent can aid analysts in gaining insight into how changes on the battlefield may affect the outcome.

2. Usability Study

Another method used to test the usefulness of GENAgent was a usability study against MANA and Archimedes. Subjects were asked to watch the building of a scenario, watch a simulation run, and explore the options of the modeling tools. MANA and Archimedes were demonstrated first and then the participants were shown GENAgent. After the last model was demonstrated, the subjects were asked a series of questions comparing GENAgent to the other models individually. In the comparison, the choices were “worse”, “same”, or “better.” The GUI interfaces, ease to build, terrain, different units and different operations representation, aesthetic appeal, agent behavior, observation of emergent behavior, and statistical capability were compared.

Nine participants took part in the study. All were male military officers with an average of seven years of service. In the demographic questionnaire the subjects rated themselves on average as intermediates with infantry tactics and Multi-Agent Systems, and experts in understanding of Windows interfaces.

The results of the study had the participants finding that GENAgent on average was “better” than MANA and Archimedes in its GUI interface and statistical capability. GENAgent was found to be also on average easier to build than Archimedes. All other areas produced no over whelming indication for a “worse” or a “better” opinion. With all these “same” averages and the few “better” areas, it can be concluded that GENAgent is equaled to or slightly improved upon the two existing simulations. Appendix F is a table of average scores and questions included in the study.

B. FUTURE WORK

For the future work suggestions, we focus on GENAgent simulation. MANA and Archimedes are in beta phases and will not be addressed. GENAgent is the next generations of GI Agent, enhancing its capabilities in the sense of modeling agent based combat simulation with the application of a generalization framework. The organizational and software structures of GENAgent are basically based on GI Agent. Some future work suggestions are related to inherited properties.

1. Improving Agent Characteristics

In GENAgent, agents have tangible (speed, sensing range, shooting range etc.) and intangible (obedience, training level, loyalty, etc.) characteristics. In real life, these are not stand-alone characteristics. They have effects on each other. For example an agent being tired may affect its speed and probability of hit. Agent’s personality interacts with its capabilities. GENAgent is just in a developing stage of MAS entity-level combat simulations. Realistic and detailed relationships among tangible and intangible characteristics can be added and this would improve the simulation’s usefulness.

2. Agent Capabilities

GENAgent's agents can visually sense the environment, walk, shoot and communicate. Some future work might include adding more realistic characteristics, adding genetic algorithms in agent decision making so an agent can learn from its mistakes and improve its performance over time.

3. Improving Simulation Capabilities

GENAgent is currently capable of handle up to a company level of homogenous forces for each side. An improvement would be to have multiple units for each side with different operation points and characteristics. For example, blue could have a platoon to set an ambush and a company to defend a position. Also, the ticket implementation for operational orders can be improved to read the order from a text file in a certain format. This would eliminate the need to reset parameters for each simulation. Another point may be to give the simulation optimization capability on a given parameter.

4. Realistic Weapons & Weapon Selection

GENAgent agents are armed with a standard weapon system whose lethality level can be adjusted. Weapon effects can be based on real data and agents can be given the capability to carry multiple weapons and select the most appropriate one for the current condition.

5. Operations on Realistic Terrain

GENAgent terrain consists of terrain squares as individual terrain blocks. An improvement would be to add the ability to import digital map data and have a real terrain map and real terrain features for the agents to interact with. For example terrain map can be imported from DTED format.

6. Summary of Goals

By developing the two scenarios in our "test" models we were able to model a variety of combat entities and combat operations using one framework. The framework helped create MANA and Archimedes simulations by easily converting a complex scenario into the essential combat components. The development of GENAgent created a simulation laboratory that utilizes this framework and quickly focuses a user on the important parts of a combat scenario needed in a MAS. The analysis of GENAgent's scenario outputs provided validation of its ability to produce realistic and reasonable data.

C. CONCLUSION

In a MAS all of the details from a scenario are not included. In this type of distillation simulation only what is "of essence" is modeled. An advantage is that the analysts can investigate the "behavioral" dynamics in a simulation and not the physics of weapon ballistics or validity of probability of kill tables (Lauren, 2001). As the popularity of MAS and distillation simulations increase and their usefulness is realized, there becomes a need to think about combat entities and combat operations in a generalized/ non-specific way. Many of the details from a real life combat scenario can be thrown out and the heart of what is being analyzed is left. The generalization framework provides this capability.

The generalization framework provides the minimum values and functionality that is needed for a robust MAS combat simulation. It allows for the ability to include many different combat units or entities and have a wide range of operations for the entities to perform in the simulation. The usefulness of this framework was shown in two existing modeling tools and in the development of GENAgent.

GENAgent is the next generation in the ongoing agent-based work at the Naval Postgraduate School. There are still functions that need to be implemented, but in its present beta form it is a robust agent based modeling tool that provides the ability to capture the adaptability and other "emerging behavior" in combat.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. OFFENSIVE AND DEFENSIVE OPERATIONS FROM U.S. ARMY FM 3-0 (2001)

<table>
<tr><th colspan="4">OFFENSIVE OPERATIONS</th></tr>
<tr><th>OPERATION</th><th>DEFINITION</th><th colspan="2">CHARACTERISTICS</th></tr>
<tr><td rowspan="2">Movement to Contact</td><td rowspan="2">A type of offensive operation designed to develop the situation and establish or regain contact.</td><td rowspan="2">There are two main types of movement to contact:</td><td>Search and attack is a technique for conducting a movement to contact that shares many of the characteristics of an area security mission.</td></tr>
<tr><td>Meeting engagement is a combat action that occurs when a moving force engages an enemy at an unexpected time end place.</td></tr>
<tr><td rowspan="3">Attack</td><td rowspan="3">An offensive operation that destroys or defeats enemy forces, seizes and secures terrain, or both.</td><td rowspan="3">Types of attack:</td><td>Hasty attack</td></tr>
<tr><td>Deliberate attack</td></tr>
<tr><td>Special purpose (Spoiling, Counterattack, Raid, Ambush, Feint, Demonstration)</td></tr>
<tr><td>Exploitation</td><td>A type of offensive operation that usually follows a successful attack and is designed to disorganize the enemy in depth.</td><td colspan="2">Exploitation seeks to disintegrate enemy forces to the point where they have no alternative but surrender or flight.</td></tr>
<tr><td>Pursuit</td><td>A type of offensive operation designed to catch or cut off a hostile force attempting to escape with the aim of destroying it.</td><td colspan="2">Pursuits are decisive operations that follow successful attacks or exploitations.</td></tr>
</table>

<table>
<tr><th colspan="4">DEFENSIVE OPERATIONS</th></tr>
<tr><th>OPERATION</th><th>DEFINITION</th><th colspan="2">CHARACTERISTICS</th></tr>
<tr><td>Mobile Defense</td><td>A type of defensive operation that concentrates on the destruction or defeat of the enemy through a decisive attack by a striking force.</td><td colspan="2">Orients on destroying attacking forces by permitting the enemy to advance into a position that exposes him to counterattack.</td></tr>
<tr><td>Area Defense</td><td>A type of defensive operation that concentrates on denying enemy forces access to designated terrain for a specific time rather than destroying the enemy outright.</td><td colspan="2">Orients on retaining terrain by drawing the enemy in an interlocking series of positions and destroy him largely by fires.</td></tr>
<tr><td rowspan="3">Retrogrades</td><td rowspan="3">A type of defensive operation that involves moving friendly forces away from the enemy to gain time, preserve forces, place the enemy in unfavorable position, or avoid combat under undesirable conditions.</td><td rowspan="3">Organized movement away from the enemy. There are three forms of retrogrades.</td><td>Delay is an operation in which a force under pressure trades space for time by slowing the enemy’s momentum, and inflicting maximum damage on the enemy becoming decisively engaged.</td></tr>
<tr><td>Withdrawal is a planned operation in which a force in contact disengages from an enemy force.</td></tr>
<tr><td>Retirement is an operation in which a force not in contact with the enemy moves away from the enemy.</td></tr>
</table>

APPENDIX B. EXAMPLES OF OPERATIONS GENERALIZATION APPLICATION

	Operation Points	Action on Objective	Action upon Contact
OFFENSE	All three points are different	Make Contact	Attack
	Obj. P. and Ending P.are the same		
DEFENSE	All three points are the same	Wait & Fortify	Attack
	Obj. P. and Ending P. are the same		
MOVEMENT	Obj. P. and Ending P. are the same	End Mission	Keep Going
			Drawback
			Attack
AMBUSH	All three points are different	Wait for Contact	Drawback
RAID	All three points are different	Make Contact	Drawback

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. “AMBUSH AT DUSK “ FORCE SETTINGS FOR GENAGENT

Parameters \ Forces	BLUE	RED
Squad Elements	9	9
Platoon Elements	3	-
Force Level	Platoon	Squad
Action in Contact	Attack	Drawback
Action in Objective Point	End Mission	Wait for Contact
Break Point (%)	50	60
Waiting Time Limit	70	70
Sensing Range	6	8
Weapons Range	4	6
Prob. to Hit	0.6	0.6
Maximum Range	1	1
Durability	5	5
Lethality	1	1,2,3 (Respectively)
Training (%)	100	100
Simulation Run Mode	Data Collection with 30 repeats	

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. "ENEMY OVER BRIDGE" FORCE SETTING FOR GENAGENT

Parameters \ Forces	BLUE	RED
Squad Elements	9	9
Platoon Elements	3	3
Company Elements	2	-
Force Level	Company	Platoon
Action in Contact	Attack	Attack
Action in Objective Point	Make Contact	Wait & Fortify
Break Point (%)	60	70
Waiting Time Limit	70	70
Sensing Range	6	6
Weapons Range	4	4
Prob. to Hit	0.7	0.7
Maximum Range	1	1
Durability	5	5
Lethality	1	1
Training (%)	100,80,60	100
Simulation Run Mode	Data Collection with 30 repeats	

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. MODELING TOOLS COMPARISON

General Categories	Specific Categories	GENAgent	MANA	Archimedes
Software & Simulation Development	Time to Develop Software	One person, part time work, 3 mos.	One person, part time work, 1 year	Two-people team, full time, 6 mos.
	Complexity of Developing Scenario	Simple	Medium	Difficult
	Providing GUI to Develop Scenario	Yes	Yes	Yes/API
	Ease in Using GUI	Easy	Easy	Medium
Simulation Capabilities	Multiple Runs	Yes	Yes	Yes
	Terrain Creation	Simple	Simple	Simple
	Different Terrain Features	Yes	Limited	Yes
	Providing Terrain Information	Yes (For selected square)	No	Limited
	Data Collection	Yes	Limited	Yes
	Ease in Setting Simulation Parameters	Easy	Easy	Medium
	Realistic Terrain Appearance	No	Yes	No
Agent & Force Capabilities	Representing Multiple Forces	No	Yes	Yes
	GUI for Setting Agent Characteristics	Yes	Yes	Yes/API
	Run-Time Agent Status	Yes	No	No
	Agent Brain Lid	Yes	No	No
	Representing Individual Agents	Yes	Yes	Yes
	Force Aggregation Capability	No	Yes	Yes

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F. USABILITY STUDY RESULTS

Each question asked the participants to compare a property of GENAgent to MANA or Archimedes. There were three choices “worse”, “same”, and “better”. A value of 1, 2, or 3, respectively, was assigned to the choices.

The results are below. A value greater than 2.5 is considered ‘better”. Values between 1.5 and 2.5 were considered “same” and below 1.5 is considered “worse”.

Question	Compared to MANA	Compared to Archimedes
GUI Interface	2.89 Better	2.89 Better
Simplicity to Build Scenario	2.56 Better	2.67 Better
Terrain	2.56 Better	2 Same
Combat Unit Representation	2 Same	2.33 Same
Combat Operation Representation	2.56 Better	2.44 Same
Aesthetic Appeal	1.78 Same	2.22 Same
Realistic Agent Behavior	2.56 Better	2.56 Better
Ability to Observe Emergent Behavior	2.55 Same	2.44 Same
Data Collection Capability	2.89 Better	2.89 Better

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G. INSTALLING AND RUNNING GENAGENT

The following instructions are intended to give the user specific instructions to install and run GENAgent.

1. Check to see if you have the latest Java build on your machine.
   a. At the "C prompt" type: **java –version**. You should see something like:

      JAVA VERSION "1.3.1"

      Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1-C)

      Java HotSpot(TM) Client VM (build 1.3.0-C, mixed mode)

      The version should be "1.2.0" or higher.
2. If the latest Java JDK is not installed on the computer being used:
   a. Copy the "j2sdk1_3_1-win.exe" file to the computers desktop, or other temporary directory.
   b. Double click the icon to start installation, or run the ".exe" file. Java version 1.3.1 will be installed and set up on your machine.
3. Copy the folder: "GENAgent" into your computer.
4. To run the GENAgent simulation:
   a. Open a DOS window and move into GENAgent directory
   b. At the command prompt type "java AgentSim".
   c. If you have problems in running GENAgent in DOS environment because of some missing library files, you could run the program in Jbuilder 4.0 or higher.
   d. If you don't have Jbuilder 4.0, you can download it from Borland's web page.

e. Run Jbuilder, and open "generailation.jpr" project file in GENAgent folder.
f. Run the project using the GUI.

For the simulation developer: The complete code is available in the CD attached and in the web page http://www.npsnet.org/~moves/. We encourage any interested parties to look through the code and if there are any questions or comments, please contact esremer@yahoo.com

LIST OF REFERENCES

Anderson, Mark, *MANA Combat Model Help File*, Defense Operation Technology Support Establishment (DOTSE), New Zealand, 2001.

Axelrod, Robert, *The Complexity of Cooperation*, Princeton University Press, Princeton, New Jersey, 1997.

Bassford, Christopher, *Doctrinal Complexity: Non-linearity in Marine Corps Doctrine*, Maneuver Warfare Science 1998, Washington, D. C., 1998.

Bohman, William E., *STAFFSIM, An Interactive Simulation for Rapid, Real Time Course of Action Analysis by U.S. Army Brigade Staffs*, Master's thesis, Naval Postgraduate School, Monterey, CA, 1999.

Clausewitz, Karl Von, *On War*, Trans. O.J. Matthijs Jolles, Random House, 1943.

Davis, Paul, K., *An Introduction to Variable-Resolution Modeling and Cross Resolution Model Connection*, RAND And RAND Graduate School of Policy Studies, RAND Report No. R-4252-DARPA , 1993.

Ferber, Jacques, *Multiagent Systems- An Introduction to Distributed Artificial Intelligence*, Pearson Education Limited, Great Britain, 1999.

Ilachinsky, Andrew, *Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An Artificial Life Approach to Land Combat*, Center of Naval Analyses, 1996.

Hartman, J.K., Parry, S.H., Caldwell, W.J., *Aggregated Combat Modeling*, Operations Research Department, Naval Postgraduate School.

Headquarters, Department of the Army, Operations (Field Manual 3-0), Washington, DC: U.S. Government Printing Office, 2001.

Headquarters, Department of the Army, *Infantry Rifle Platoon and Squad* (Field Manual 7-8), Washington, DC: U.S. Government Printing Office, 1992.

Hiles, J., *Course Notes for MV-4015 Agent-Based Autonomous Behavior for Simulations*. Summer, 2001, Naval Postgraduate School.

Lanchester, F.W., *Aircraft In Warfare*, Lanchester Press (Originally published in 1916 by Constable & Co.) London, England, 1995.

Lauren, Michael K., *Applications of a Distillation to Questions For the NZ Army*, Maneuver Warfare Science 2001, Marine Corps Combat Development Command, Quantico VA, 2001.

Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). *The Swarm Simulation system: A Toolkit for Building Multi-Agent Simulation*s. http://www.swarm.org/intro.html (30 Jul 99).

Operations Research Department, *Aggregated Combat Models*, Naval Postgraduate School, Monterey Ca., 2000.

Pawloski, Joel S., *Modeling Tactical Land Combat Using a Multi-Agent System Design Paradigm (GI Agent),* Master's thesis, Naval Postgraduate School, Monterey, CA, 2001.

Reynolds, William N. and Dixon, David S., *Archimedes: A Prototype Distillation*, Maneuver Warfare Science, 2001.

Roddy, Kimberly A. and Dickson, Michael R., *Modeling Human And Organizational Behavior Using A Relation-Centric Multi-Agent System Design Paradigm*, Master's thesis, Naval Postgraduate School, Monterey, CA, 2000.

Rosenbloom, Laird, and Newell, *The Soar Papers: Readings on Integrated Intelligence*, 1993

Science Applications International Corporation (SACI) and Lockheed Martin Information Systems, *MODSAF 5.0: FUNCTIONAL DESCRIPTION DOCUMENT (FDD)*, Advanced Distributed Simulation Technology II, CDLR, Orlando, FL, 1999.

Schmitt, Maj. John F., USMCR, *Marine Corps Gazette's Mastering Tactics – A Tactical Decision Games Workbook*, Marine Corps Association, Quantico VA, 1993.

Stine, J., *Representing Tactical Land Navigation Expertise*, Master's thesis, Naval Postgraduate School, Monterey, CA, 2000.

U.S. Army Training and Doctrine Command Analysis Center (TRAC), JANUS version 7.06D, [Computer Software], White Sands Missile Range, NM, 1999.

Unrath, Craig S., *Dynamic Exploration of Helicopter Reconnaissance Through Agent-based Modeling*, Master's thesis, Naval Postgraduate School, Monterey, CA, 2000.

Upton, Stephen C., *Warfare and Complexity Theory*, A Primer, Maneuver Warfare Science 1998, Washington, D.C., 1998.

Weiss, Gerhard, Multiagent Systems-A Modern Approach to Distributed Artificial Intelligence, Massachusetts Institute of Technology, 1999

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8735 John J. Kingman Rd, STE 0944
   Fort Belvoir, Virginia 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd
   Monterey, California 93943-5101

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California
   debarber@nps.navy.mil

4. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia
   webmaster@tecom.usmc.mil

5. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia
   ramkeyce@tecom.usmc.mil
   strongka@tecom.usmc.mil
   sanftlebenka@tecom.usmc.mil

6. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California
   doranfv@mctssa.usmc.mil
   palanaj@mctssa.usmc.mil

7. Director, Studies and Analysis Division, MCCDC, Code C45
   Quantico, Virginia
   thesis@mccdc.usmc.mil

8. Kara Harp Okulu Komutanligi (Turkish Army Academy)
   Bakanliklar, Ankara / Turkey
   akdan@kho.edu.tr

9. Harp Akademiler Komutanligi
   4. Levent, Istanbul / Turkey
   ktp@harpak.tsk.mil.tr

10. S. Bugra KARAHAN
4. Levent, Istanbul / Turkey
bkarahan@harpak.tsk.mil.tr

11. Middle East Technical University
Balgat, Ankara / Turkey
lib-hot-line@metu.edu.tr

12. Bilkent University
Balgat, Ankara / Turkey
library@bilkent.edu.tr

13. Esref MERT
4. Levent, Istanbul / Turkey
esremer@yahoo.com

14. Erik Jilson
Quantico, Virginia
eriknwendy@earthlink.net

15. Dr. Michael J. Zyda
Chair, Modeling, Virtual Environments and Simulation (MOVES)
Naval Postgraduate School
Monterey, California
zyda@movesinstitute.org

16. Dr. Rudolph P. Darken
MOVES Academic Associate
Naval Postgraduate School
Monterey, California
darken@nps.navy.mil

17. Dr. Donald Gaver
Operations Research Department
Naval Postgraduate School
Monterey, California
dgaver@nps.navy.mil

18. Mr. John Hiles
MOVES Academic Group
Naval Postgraduate School
Monterey, California
jhiles@mindspring.com

19. Maj. Michael Van Putte
Naval Postgraduate School

Monterey, California
mavanput@nps.navy.mil

20. Curtis Blais
Naval Postgraduate School
Monterey, California
clblais@nps.navy.mil

21. Dr. Gary E. Horne
MITRE Quntico Principle Scientist
Quantico, Virginia
hornege@mccdc.usmc.mil